Machine Learning Techniques for High Dimensional Data

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy
by
Yuan Chi

October 2015

Abstract

This thesis presents data processing techniques for three different but related application areas: embedding learning for classification, fusion of low bit depth images and 3D reconstruction from 2D images.

For embedding learning for classification, a novel manifold embedding method is proposed for the automated processing of large, varied data sets. The method is based on binary classification, where the embeddings are constructed so as to determine one or more unique features for each class individually from a given dataset. The proposed method is applied to examples of multiclass classification that are relevant for large scale data processing for surveillance (e.g. face recognition), where the aim is to augment decision making by reducing extremely large sets of data to a manageable level before displaying the selected subset of data to a human operator. In addition, an indicator for a weighted pairwise constraint is proposed to balance the contributions from different classes to the final optimisation, in order to better control the relative positions between the important data samples from either the same class (intraclass) or different classes (interclass). The effectiveness of the proposed method is evaluated through comparison with seven existing techniques for embedding learning, using four established databases of faces, consisting of various poses, lighting conditions and facial expressions, as well as two standard text datasets. The proposed method performs better than these existing techniques, especially for cases with small sets of training data samples.

For fusion of low bit depth images, using low bit depth images instead of full images offers a number of advantages for aerial imaging with UAVs, where there is a limited transmission rate/bandwidth. For example, reducing the need for data transmission, removing superfluous details, and reducing computational loading of on-board platforms (especially for small or micro-scale UAVs). The main drawback of using low bit depth imagery is discarding image details of the scene. Fortunately, this can be reconstructed by fusing a sequence of related low bit depth images, which have been properly aligned. To reduce computational complexity and obtain a less distorted result, a similarity transformation is used to approximate the geometric alignment between two images of the same scene. The transformation is estimated using a phase correlation technique. It is shown that that the phase correlation method is capable of registering low bit depth images, without any modification, or any pre and/or post-processing.

For 3D reconstruction from 2D images, a method is proposed to deal with the dense reconstruction after a sparse reconstruction (i.e. a sparse 3D point cloud) has been created employing the structure from motion technique. Instead of generating a dense 3D point cloud, this proposed method forms a triangle by three points in the sparse point cloud, and then maps the corresponding components in the 2D images back to the point cloud. Compared to the existing methods that use a similar approach, this method reduces the computational cost. Instated of utilising every triangle in the 3D space to do the mapping from 2D to 3D, it uses a large triangle to replace a number of small triangles for flat and almost flat areas. Compared to the reconstruction result obtained by existing techniques that aim to generate a dense point cloud, the proposed method can achieve a better result while the computational cost is comparable.

Contents

List of Figures

Acknowledgement

First of all, I am extremely grateful to my primary supervisor, Prof. Jason Ralph, for his support, advice and encouragement throughout the whole duration of my PhD study. His deep insights helped me at various stages of my research, especially the useful guidance during the difficult conceptual development stage. I also remain indebted for his financial support. My sincere gratitude also goes to my co-supervisor, Dr. Yannis Goulermas, for his invaluable insights and suggestions. I really appreciate his willingness to meet me at short notice every time.

I wish to express my sincere appreciation to the other colleagues and friends who I met during my time at the University of Liverpool, and those who have contributed to this thesis and supported me in one way or the other. Special thanks here are due to our postdoctoral researcher, Dr. Elias Griffith, whose help with MATLAB programming was of utmost importance when I was in the early stages of my PhD study. I would also like to take this opportunity to thank Dr. Tingting Mu and Prof. Stephen Marshall - my viva examiners, for their very helpful comments and suggestions.

Last but not least, words cannot express the feelings that I have for my parents for their constant unconditional support over the years - both emotionally and financially. I would not be where I am today if it not for them, and for which I could never thank them enough.

Chapter 1

Introduction

1.1 Research Topics

There are three applications (i.e. embedding learning, image fusion, and 3D reconstruction) considered in this thesis, but there are two main areas of work: embedding learning and image registration. Embedding learning techniques aim to convert data from a high dimensional representation to a lower one while preserving the intrinsic geometry of the data, while image registration techniques aim to determine a geometrical transformation that aligns two images of the same scene. Although they do not seem directly related to each other at first, both of them are often included as processing stages in applications of computer vision and other relevant areas of science and engineering. More recently, embedding learning is used as a pre-processing stage to enhance the performance of image registration, such as for medical imaging [1–3], and in hyperspectral imaging [4, 5].

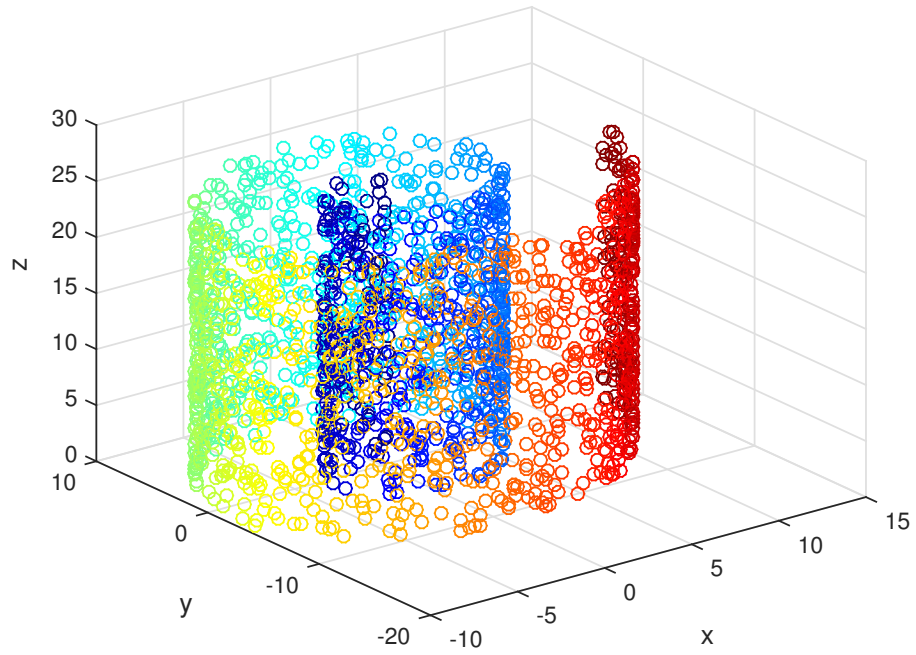
1.1.1 Embedding Learning for Classification

Handling measurements in a high dimensional mathematical space occurs frequently in applications of machine learning, computer vision, data compression, and information retrieval, as well as many other areas of science and engineering. The measurements are often represented as feature vectors, and the vector space associated with these vectors is called the feature space. For example, a statistical classification task commonly refers to determining which of a set of categories (i.e. classes) one or more new measurements (i.e. data samples) should belong to, on the basis of an existing set of measurements whose corresponding category information (i.e. labels) is known. Many classical classification algorithms have been proposed, and they are widely employed, e.g. Neural Networks (NNs) [6,7], Support Vector Machines (SVMs) [8,9] and k-Nearest Neighbour (k-NN) classifiers [10, 11]. In real-world applications, data sets are usually represented in high dimensional space. For example, a face image captured from a digital image or a video frame often has many more degrees of freedom than that is required to define and/or determine a perceptually meaningful structure for the purpose of identification

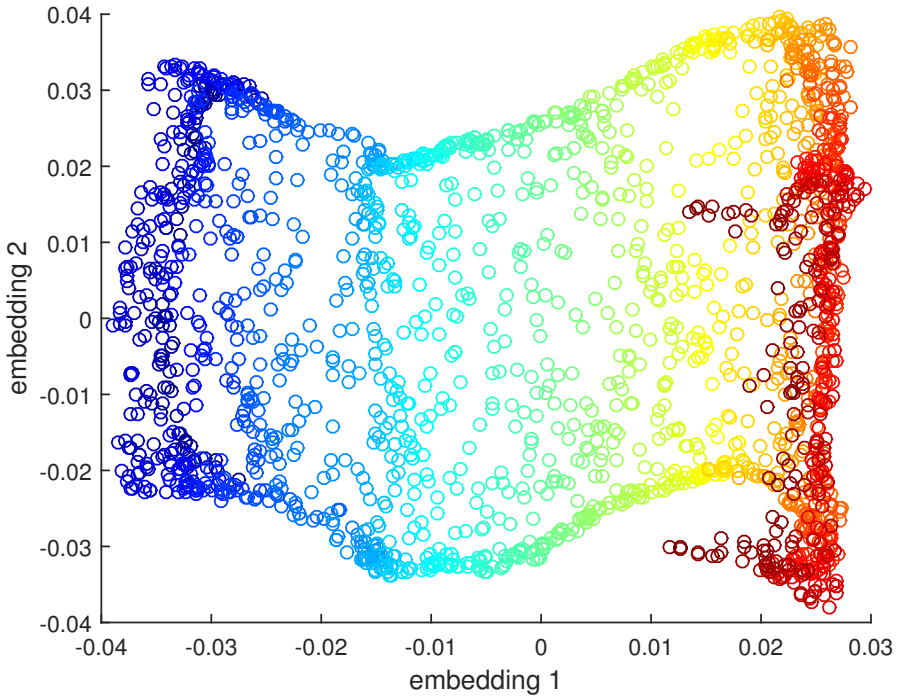
or verification [12, 13]. Classifiers often suffer from some major problems as data dimension increases, such as increasing computational complexity, producing extremely unreliable final results and so on [14, 15]. To overcome these problems, techniques that aim to learn a *compact* representation of the original multivariate data sets are often employed. These reduce dimensionality by eliminating irrelevant and/or redundant information that is present in the original high dimensional data.

In general, embedding learning refers to the process of extracting a subset from a set of data samples. For statistical classification, embedding learning can provide a compact and meaningful representation for a set of data samples in a high dimensional feature space by defining or determining a specific embedded subset of data samples in a reasonably low dimensional feature space from the original (high dimensional) set. The resulting embedded set should preserve some of the original properties and characteristics of the data. An example of learning a specific set of 2D embeddings from a set of 3D data samples is shown in Figure 1.1 [13], which was plotted by using MATLAB 2014b. The manifold of the data set (i.e. the local neighbour structures) that is present in the 3D space is unwrapped, as it is converted to a 2D space.

In the literature, a broad range of algorithms for embedding learning have been proposed [15, 16]. Most algorithms differ from each other in the way that they preserve important properties and characteristics that are present in the original set of data samples during the learning process. The learning process refers to solving an optimisation problem, which seeks to minimise or maximise an objective function, and whose solution must satisfy a certain condition (i.e. a constraint). To achieve different preservations of the important properties and characteristics, different objective functions and the corresponding constraints are imposed according to the existing techniques. Given a set of data samples that may have corresponding label information (i.e. which class or classes each data sample belongs to), existing methods can be divided into the following three categories, based on whether to utilise the label information and what type of the label information is available. The first category is unsupervised learning, where the techniques compute embeddings based only on certain feature properties and characteristics that exist in the original data set, regardless of whether the corresponding label information exists. The embedded set learned by unsupervised techniques is a compact and informative representation of the original set which is based only on features. The second category is supervised learning techniques, where the techniques utilise the feature data and the corresponding label information when embeddings are being constructed. The resultant supervised embeddings outperform unsupervised ones in two ways: one is the grouping ability of the ones in the same class (i.e. keeping interclass ones close together), and the other is the separation ability of the ones from different classes (i.e. forcing interclass ones to move far apart). The last category is called semi-supervised learning techniques, which are developed for the situation where



(a) A set of 3D data samples



(b) An embedded set of 2D data samples

Figure 1.1: An example of embedding learning (taken from [13] but redrawn here): representing a set of 3D data samples in a 2D feature space, while preserving the local neighbour structure of each data sample in the original 3D feature space.

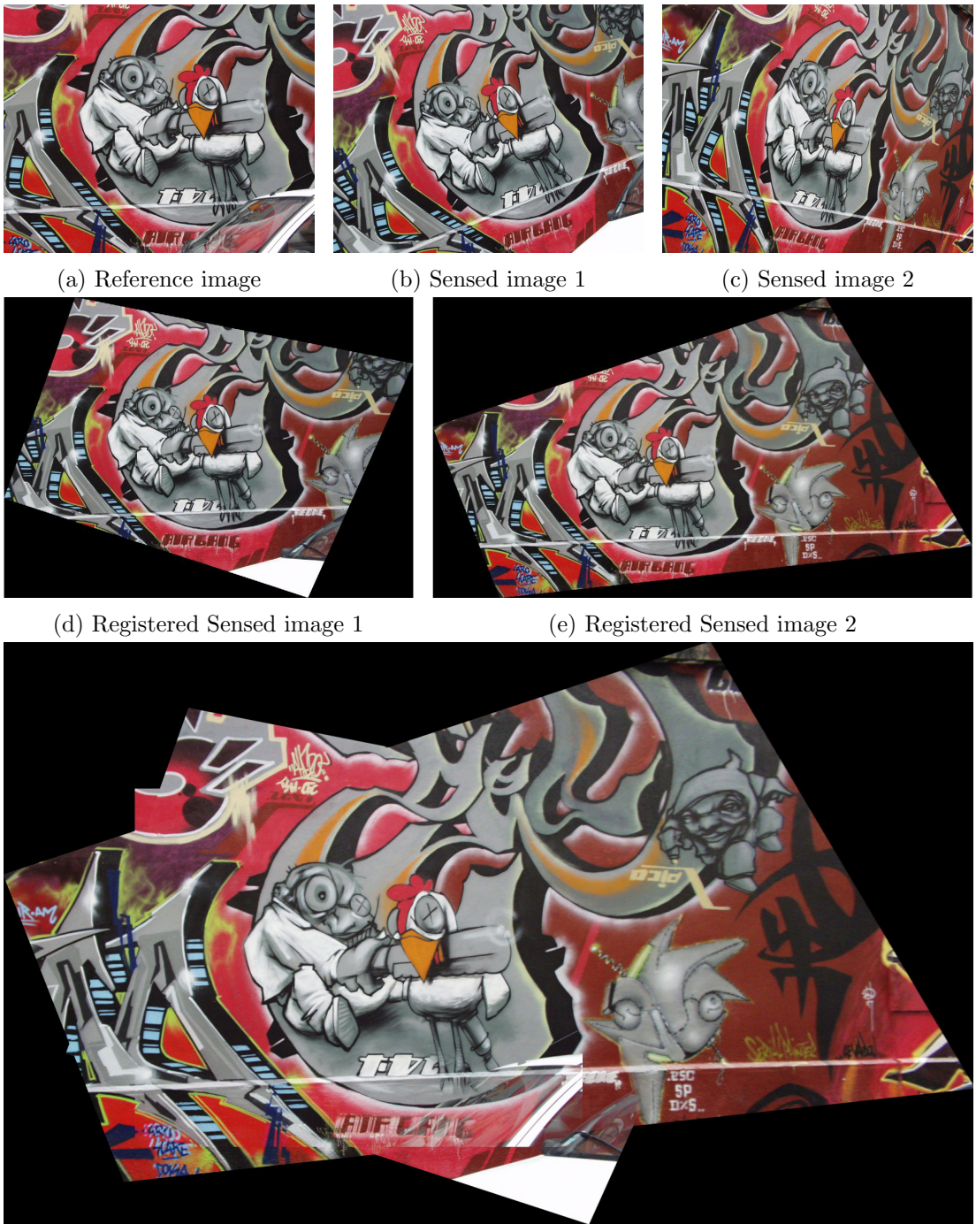
the label information is not complete (i.e. only part of the data set is labelled). Most of these algorithms are some kind of combination of an unsupervised learning method and a supervised one, i.e. employing an unsupervised learning method to consider all the data samples and a supervised one to deal with only the labelled data samples. The final performance of semi-supervised embedding learning methods are enhanced compared to unsupervised learning, due to the employment of the supervised learning techniques and the use of additional information (labels).

1.1.2 Image Registration & Its Application in 3D Reconstruction

Image registration is a fundamental problem in applications of image analysis related areas. It aims to find one or more spatial transformations, so that two or more images (commonly one reference image with one or more sensed images) of the same scene can be geometrically aligned. Although the images are of the same scene, they differ from each other because they may be taken at different times, from different viewpoints, and/or by different sensors [17, 18]. For instance, in medicine, images taken from different modalities (i.e. using different wave bands, such as X-ray radiography, Magnetic Resonance Imaging (MRI), medical ultrasonography and so on) are aligned to assist diagnosis. Image registration has become a crucial stage in tasks such as remote sensing, medical imaging and computer vision, where final information is based on combining a number of images. An example of image registration for image mosaicing [19] is shown in Figure 1.2.

Generally, in 2D images, there are three major types of variations in images that make them look distinct [17]. The first type of variation is caused by the process of image acquisition, which leads to image misalignment. The second variations are changes in the intensity values, due to changes in lighting conditions or relative contrast in different modalities. The last type of variation is caused by changes in the scene, such as moving objects. A spatial transformation between two images is sufficient to remove the first type of variation, but not the other two types of variation. They generate a more difficult problem in image registration, since an exact match is difficult to achieve [17]. The problem would not be critical, as long as the changes in intensity values are relatively small compared to the original intensity values, and the moving object is not the object of interest in the taken images. As shown in Figure 1.2d, the part of a car in the bottom right of the image (white area) is not removed after applying image registration. Also as shown in Figure 1.2f, the final result of image mosaicing is reasonable, because the part of the car is not the main object of interest in the three images.

Over the years, a broad range of techniques for image registration have been proposed for a wide variety of areas [17, 18, 20, 21]. Unfortunately, because images to be registered are normally acquired under a large variety of different conditions, there is



(a) Reference image (b) Sensed image 1 (c) Sensed image 2

(d) Registered Sensed image 1 (e) Registered Sensed image 2

(f) Final result of image mosaicing

Figure 1.2: An example of image registration for the application area in image mosaicing

no one existing method that can achieve an optimal performance when the alignment is across different types of sensor, platform and/or problem [18, 22]. Existing techniques for image registration can be classified into two major categories: area based methods and feature based ones. The area based methods process images without detecting any salient points, but consider image areas as features. Usually, windows (i.e. an area of

the image) of predefined size are used for the estimation of corresponding transformation models, which describes mathematical relationships between images for alignment. However, because they only make direct use of matching image intensity values, with no analysis of image structures, they are sensitive to the changes in intensity value. Compared to area based methods, feature based ones work by detecting more robust features: usually sets of points or small blocks detected in the reference and sensed images. The aim of these methods is to find the pairwise correspondence between the detected features in the different images, using either the spatial relations or various feature descriptors.

In general, the process of image registration can be divided into the following four stages: detecting features, where the information in the images is useful for matching is extracted; matching features, where the correspondences are established between the features detected in the reference and sensed images; estimating the transformation model, where the type and the parameters in the spatial transformation are estimated, on the basis of the established correspondences of the detected features; and transforming & resampling images, where the sensed images are transformed and resampled according to the estimates of the spatial transformation [18].

Detecting Features Features commonly refer to significant areas in images, i.e. salient regions, lines or even points, which are manually selected or, preferably, automatically detected [17,18]. Nevertheless, the features should be distinctive, expected to be easily detectable and somehow stable, i.e. the same features detected from different images should share sufficient common elements, even in situations where images do not cover the exactly the same scene and/or the objects present in images are occluded. Ideally, any applicable method should be able to detect the same features in all of the possible images of the same scene, and be insensitive to any assumed type of geometric deformation and/or additive noise.

Matching Features In this stage, the features detected in different images are matched to each other. The matching process is based on either various measures of image intensity values between the detected features and the several nearest neighbours, or a similarity measure of the descriptors constructed according to the detected features. Since any incorrectly matched pairs of features will cause the performance to suffer in the following stages, the algorithms proposed for matching features should be robust. Some existing techniques (i.e. area based methods) merge the current stage with the next one, because once the pairs of feature correspondences have been determined, they can directly estimate the parameters required by the transformation model.

Estimating Transformation Models The spatial transformation that can geometrically align two images is estimated in this stage. Based on the prior information about the assumed transformation model, the parameters required by the spatial transformation are computed using the established matched pairs of feature correspondences. The estimated model of the spatial transformation should result in a reasonable transformation, where the corresponding matched pairs are located as close as possible. If there is no prior information available, the estimated model should be both flexible and general enough to handle all possible spatial transformations for geometrical alignment.

Transforming & Resampling Images After the model of a spatial transformation has been estimated, the images can be geometrically aligned by mapping the sensed image(s) into the spatial coordinate of the reference image. Since the new pixel coordinate of the registered image is determined according to that of the reference image, a non-integer pixel location is often generated, where no appropriate intensity value can be assigned directly. To address this problem, the sensed images are often resampled employing an appropriate intensity interpolation technique. The choice of the intensity interpolation technique depends on the trade-off between the registration accuracy and the computational complexity. In general, the technique of bilinear interpolation is sufficient enough for practical applications [18].

3D Reconstruction from 2D Images

3D reconstruction from 2D images is a challenging task in computer vision and computer graphics. It represents a process of obtaining 3D information about the geometry of a scene from 2D images of this scene. Depth data is the 3D component missing from given 2D images. To recover depth data, image registration serves as the key part. This is because a 3D point can be reconstructed by triangulation [23], with the use of its corresponding pixel locations in the 2D images that have already been matched through image registration. To apply triangulation methods, it is an important prerequisite that the pose and calibration of each camera used should be determined. Traditional methods require a priori information about 3D positions, such as the 3D location and pose of camera, or the 3D location of ground control points [23]. However, there is a method called Structure from Motion (SfM) [23, 24], which can simultaneously and automatically get the camera pose and scene geometry. It is achieved by employing a method called bundle adjustment [25, 26], which is based on matched pixel locations extracted from multiple 2D images that are overlapping and offset.

1.2 Novel Contributions

For embedding learning, a new approach has been proposed to try to overcome the problem where only a small number of data samples are available to characterise the

underlying structure of a given dataset. It attempts to generate a feature space formed by distinct feature(s) of each class, which could potentially enhance the grouping of the data samples in this class while separating them from those in the remaining classes. Compared to the existing techniques that try to find distinguishing features, each of which attempts to separate all classes, this approach focuses on finding the distinct characteristics of one class at a time. It results in the following two contributions: Firstly, the idea of One-Vs-All (OVA) [27, 28] which decomposes a multiclass classification problem into several binary ones is applied to embedding learning. It finds distinct feature(s) for each class individually, so that the data samples in this class can be uniquely defined, and then separated from those in the remaining ones. Secondly, to overcome the problem of the imbalanced number of data samples when addressing final optimisation, a weighted pairwise constraint indicator is proposed to balance the contributions of the data samples from a target class and those from the remaining classes.

For image registration, to reduce the usage of on-board computational resources in small Unmanned Aircraft System (UAS) platforms, an approach can be employed using low bit depth images instead of full bit depth ones. It only transmits the first few Most Significant Bits (MSBs) of image data, and it is based on the assumption that the most important information of each image (e.g. the main structure of the scene) is contained in the MSBs. To address the problem that low bit depth imagery can discard image details within the Least Significant Bits (LSBs), it has been found that the discarded details may be reconstructed by fusing a sequence of related low bit depth images if there is sufficient noise in the images, when the temporal sequence has been aligned by image registration [29]. Additionally, the area based registration method Phase (Fourier) Correlation is employed here, to achieve a low computational complexity. It results of the following two contributions: Firstly, the phase correlation method is demonstrated to be capable of registering low bit depth images (up to a distortion caused by a similarity transformation), without any modification, or any pre and/or post-processing. Secondly, the image details discarded due to quantisation of low bit depth imagery can be reconstructed with high fidelity by fusing a number of related low bit depth images with noise.

For 3D reconstruction from 2D images, a new approach has been proposed to achieve a good trade-off between the reconstruction details and the computational load. Unlike the common approach that maps 2D image triangles one by one to the 3D scene, it forms several large triangles to describe a flat or almost flat region. Each of the large triangles actually covers a number of small triangles, and they are formed based on the boundary points of the region. It results in two contributions. Firstly, to describe a region with a number of triangles, the triangles formed based on boundary points of this region can achieve a minimum solution. Secondly, for a flat or almost flat region, several large

triangles can achieve almost same performances when describing this region, compared one that based on all small triangles at considerably lower computational cost.

1.2.1 Publications

Parts of the above novel contributions have been presented at conferences and published in a journal paper:

1. Y. Chi, E.J. Griffith, and J.F. Ralph. Low bit depth images for small, micro-scale UAVs. In *6th International Conference on Imaging for Crime Detection and Prevention (ICDP 2015), London, UK, 15-17 Jul. 2015*, pages 21–26. IET, 2015

2. Y. Chi, E.J. Griffith, J.Y. Goulermas, and J.F. Ralph. Binary data embedding framework for multiclass classification. *IEEE Trans. Human-Mach. Syst.*, 45(4):453–464, 2015

3. Y. Chi, E.J. Griffith, J.Y. Goulermas, and J.F. Ralph. Binary data embedding framework for face recognition. In *5th International Conference on Imaging for Crime Detection and Prevention (ICDP 2013), London, UK, 16-17 Dec. 2013*, pages 102–107. IET, 2013

4. E.J. Griffith, Y. Chi, M. Jump, and J.F. Ralph. Equivalence of BRISK descriptors for the registration of variable bit-depth aerial imagery. In *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Manchester, UK, 13-16 Oct. 2013*, pages 2587–2592. IEEE, 2013

1.3 Structure of Thesis

This thesis is further divided into six chapters:

- Chapter 2, **Embedding Learning for Classification**, describes the main concepts of embedding learning for classification, and examines the commonly used techniques in unsupervised, supervised and semi-supervised embedding learning, as well as the embedding learning techniques for multi-label classification.

- Chapter 3, **Image Registration & Its Application in 3D Reconstruction**, shows how a pinhole camera model leads two 2D images of the same scene to be related by a transformation model, and examines the commonly used techniques for registering 2D images - both area based and feature based. Additionally, the structure from motion technique is examined here, it reconstructs a 3D scene from a number of 2D images of this scene based on image registration.

- Chapter 4, **Binary Data Embedding Framework**, describes the motivation and technical details of the work that has been done for embedding learning, and additionally an experimental evaluation is conducted to demonstrate the performance of the proposed method.

- Chapter 5, **Image Reconstruction by Fusing Low Bit Depth Image**, provides the motivation and whole process of the work that has been done for image reconstruction, as well as an evaluation of the reconstruction process to show the reconstruction quality.

- Chapter 6, **3D Scene Reconstruction from 2D images**, presents the motivation and whole process of the work done that has been done for 3D reconstruction, and additionally an experimental evaluation is conducted to demonstrate the performance of the proposed method.

- Chapter 7, **Conclusions & Future Work**, summarises the overall work that has been done, and suggests possible future work.

Chapter 2

Embedding Learning for Classification

This chapter, describes the background of embedding learning and introduces the standard mathematical notation that will be used in later chapters. Subsequently, a broad review of the well known techniques for embedding learning is presented.

2.1 Background

Embedding learning aims to embed a set of data samples into a lower dimensional feature space that can capture the meaningful degrees of freedom present in the data. It has become an essential processing stage in many applications in science and engineering [14]. In recent years, a wide range of algorithms for embedding learning have been proposed [15, 16]. The existing algorithms can be split into two different categories, based on whether there is an explicit relation between a set of data samples and its corresponding set of embeddings. One is nonlinear learning, where the techniques, such as Locally Linear Embedding (LLE) [13], Laplacian Eigenmaps (LE) [34], and other algorithms proposed in [12, 35–40], generate resultant embeddings that have no explicit relation to the original set. In other words, there is no linear relationship between the original set and the corresponding embedding results. The other category is linear learning, where there exists an explicit relation, and it is captured by a linear mapping/projection function. Here, the focus will be on the methods in the later category, since they are usually insensitive to the parameters [41], have lower computational complexity [41], and are simpler and straightforward for real-time applications. For instance, for the purpose of classification, once a linear mapping/projection function has been determined from a training set (i.e. a set of data used for learning), it can be used for mapping/projecting any data sample in the corresponding test set (i.e. a set of data used solely for testing generalisation performance) to the same feature space, for further identification or verification.

The early techniques were proposed to work in an unsupervised way, where they

generate embeddings only based on specific feature properties and characteristics of a set of data samples, without explicit knowledge of the identifies or classes of the data. For example, Principal Component Analysis (PCA) [42] aims to yield an orthogonal projection function, so that the corresponding embedding results are uncorrelated to each other, and the corresponding total variance is maximised. Latent Semantic Indexing (LSI) [43] attempts to obtain a lower rank approximation to represent the original set. Both methods take into account the global structure of the original set when determining the mapping functions, but not the local structure. As a result, it is likely that the local structures of the resultant embeddings become distorted. To overcome this problem, various methods using manifold learning and spectral analysis have been developed to take into account the local structure information when generating embeddings [12, 13, 34–38, 40, 44–49]. For example, Locality Preserving Projections (LPP) [44], Orthogonal Locality Preserving Projections (OLPP) [45] and Orthogonal Neighbourhood Preserving Projections (ONPP) [45] focus more on the local structures of the original set to preserve the essential geometry. This can be captured by pairwise proximity information based on constructed graphs for the local neighbours of each data sample [14]. These techniques differ from each other through different assumptions on how to construct graphs of local neighbours and enforce constraints: ONPP assumes that each data sample can be represented by a linear combination of its local neighbours, while LPP and OLPP employ similarity measures to capture graphs of local neighbours. Additionally, both OLPP and ONPP indicate that each dimensionality of the resultant linear mapping/projection function should be orthonormal.

When a set of data samples possesses a good arrangement between the features and the corresponding classes (i.e. the data samples in the same class are located nearby, while those from different classes are located far away), the above unsupervised methods can obtain a good and informative representation of the original set, based only on certain feature properties and characteristics, and no local information. However, for a real world dataset, a good arrangement may not always hold. Therefore, there may exist confusing data samples in the original set (i.e. data samples from different classes are located nearby, and/or those in the same class are located far away), and then both global and local structures of the original set become unreliable due to the existence of the confusing data samples. The resultant embeddings, which are only learned on the basis of feature properties and the characteristics of the original set, may degrade the final performance. Additionally, the local neighbour information, which is utilised by some unsupervised methods, is based on one type of distance measure (metric) between each data sample and its neighbours in the original feature space. It may be influenced when there exists a bad arrangement between the features and the labels. As a result, these methods may fail to overcome the problem, and can increase the misclassification rates, e.g. neighbouring data samples near the class boundaries in the original feature

space may get projected into a wrong class in the embedded feature space.

To avoid such problems, techniques that use the set of data samples and the corresponding label information have been proposed. These techniques are supervised learning methods. They improve the grouping ability of the data samples in the same class (i.e. intraclass), whilst increasing the separation ability of those from different classes (i.e. interclass). Most of the supervised methods attempt to embed a set of data samples, so that the data samples of intraclass are made to be close together and the interclass data samples are as far apart as possible. For example, both Fisher Discriminant Analysis (FDA) [50] and Maximum Margin Criterion (MMC) [51] achieve this based on the additional use of the label information to capture the within-class and between-class scatters (i.e. the estimates of the corresponding covariance matrices) for respective intraclass and interclass information of the original set. Both methods encourage all of the data samples of the same class to stay close together and those of different classes to move apart, with a slight difference in the way of formulating each objective function - FDA utilises the ratio of the two scatters, while MMC [51] utilises the difference between the two scatters (intra- / interclass). Various other methods have been proposed [52–66], and the majority of these are based on using rules similar to that of FDA and MMC with slight variations or incorporating the label information into the pairwise proximity information modified by the unsupervised methods. For example, Marginal Fisher Analysis (MFA) [52], Discriminative Locality Alignment (DLA) [53], and Discriminant Neighbourhood Embedding (DNE) [54] encourage only the neighbouring data samples of intraclass and interclass to be nearby and far away, respectively. They share the same idea, but differ from each other in the way that they impose each objective function and constraints. MFA [52] is a variant of FDA, where it considers local neighbour information by redefining the between-class and within-class scatters, on the basis of k-NN information. DLA [53] works similar to MFA, but it employs a criterion whose form is similarly to that of MMC (a difference rather than a ratio). DNE [54] incorporates the label information into the graphs of local neighbourhoods constructed by OLPP. Local FDA (LFDA) [57] is another variant of FDA, where it keeps the neighbouring data samples of intraclass close, while for the data samples of interclass, it works more strictly - it makes all of them move apart. Repulsion OLPP (OLPP-R) [60] uses the neighbourhoods differently compared to LFDA: it keeps all of the data samples of intraclass close, while the neighbouring data samples of interclass move apart. Repulsion ONPP (ONPP-R) [60] and Discriminative ONPP (DONPP) [63] are more complex, both of them preserve the local data structure within each class separately, while encouraging the neighbouring data samples of interclass to move apart. They differ from each other through the use of different reconstruction weights. ONPP-R [60] uses reconstruction weights that are calculated for all data samples from the same class and incorporates a repulsion graph, which captures interclass

data samples that are close by in the original feature space. DONPP [63] uses reconstruction weights which are calculated based on the assumption that each data sample can be reconstructed from the remaining ones in its class.

The fully supervised learning techniques make the use of labelled data samples when determining optimal projection functions. But in some situations, labelling data samples is difficult, expensive or time consuming, so only a small number of them are actually labelled. Employing unsupervised learning techniques in such cases may generate unreliable results, while employing supervised learning techniques with only the labelled data samples may fail to discover the actually meaningful structures, due to the small sample size. To address this problem, a group of techniques, called semi-supervised learning have been developed to use both unlabelled and labelled data samples to build a better mapping function [63, 67–73]. The simplest way to approach semi-supervised learning is to combine an unsupervised technique and a supervised one, i.e. using the unsupervised technique to deal with all data samples (both unlabelled and labelled), while the supervised technique deals with the labelled data samples. For example, Semi-supervised DONPP (SDONPP) [63] combines ONPP with DONPP, SEmi-supervised LFDA (SELF) [69] combines PCA and LFDA, while Semi-Supervised FDA (SSFDA) [68] and Semi-Supervised MMC (SSMMC) [68] combine OLPP with FDA and MMC, respectively. There are other different ways to achieve semi-supervised learning, for example methods described in [71] and [72] attempt to learn predicted labels. A broader review on semi-supervised learning techniques can be found in [73–75].

The algorithms for supervised and semi-supervised learning typically only aim for single label classification, where each data sample is assigned exactly one label. For more complex classification problems, data samples are extended from single label to multi-label, i.e. a data sample simultaneously belongs to two or more classes. In such cases, the above algorithms are no longer applicable. To address this problem, a variety of algorithms for multi-label classification have been proposed [76–87]. A direct way to achieve embedding learning for multi-label classification is to find a optimal approach to balance some kind of statistical measure (e.g. covariance or correlation coefficient) of the embedding results and the corresponding multi-label information. For example, Partial Least Squares (PLS) [76, 77] attempts to maximise the covariance between the resultant embeddings and the corresponding multi-label information. Canonical Correlation Analysis (CCA) [79, 80] works similarly to PLS. Instead of employing covariance, it aims to maximise the correlation coefficient. A recent review of the existing algorithms for multi-label classification can be found in [15].

2.1.1 Definitions

The input to a method on linear embedding learning is a training set of n data samples of dimension d, where it is denoted by an $n \times d$ matrix $\mathbf{X}$, as:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T \tag{2.1}$$

where T denotes the transpose operator, and $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^T$ corresponds to the ith data samples in the training set. The n training data samples belong to c different classes, and this class information is modelled as an $n \times c$ binary matrix $\mathbf{Y}$ as:

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]^T \tag{2.2}$$

where $\mathbf{y}_i = [y_{i1}, y_{i2}, \ldots, y_{ic}]^T$ indicates which of the c different classes $(C_1, C_2, \ldots, C_c)$ that the ith training data sample belongs to, and the ijth element y_{ij} is defined as:

$$y_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \in C_j \text{ (i.e. } \mathbf{x}_i \text{ belongs to } C_j) \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

The number of the training data samples from the ith class C_i is denoted by n_i $(i = 1, \ldots, c)$. The corresponding test set consists of m data samples of dimension d, and it is denoted by a $m \times d$ matrix $\widetilde{\mathbf{X}}$, as:

$$\widetilde{\mathbf{X}} = [\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2, \ldots, \widetilde{\mathbf{x}}_m]^T \tag{2.4}$$

where $\widetilde{\mathbf{x}}_i = [\widetilde{x}_{i1}, \widetilde{x}_{i2}, \ldots, \widetilde{x}_{id}]^T$ corresponds to the ith test data sample.

The objective of the techniques for linear embedding learning is to construct a linear mapping/projection function $\psi : R^d \to R^k$ from either only the training set $\mathbf{X}$ or both the training set $\mathbf{X}$ and the corresponding label information $\mathbf{Y}$. The function ψ is denoted by a $d \times k$ matrix $\mathbf{V}$. It generates a training set of n embeddings of dimension k $(k < d)$, denoted by a $n \times k$ matrix $\mathbf{Z}$, as:

$$\mathbf{Z} = \mathbf{X}\mathbf{V} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n]^T \tag{2.5}$$

where $\mathbf{z}_i = \mathbf{V}^T\mathbf{x}_i = [z_{i1}, z_{i2}, \ldots, z_{ik}]^T$ corresponds to the ith data sample in the embedded training set. In the embedded feature space, the dimensionality of the training set is reduced, and additionally the manifolds in the low dimensional space that the training data samples lie on should be recovered, as well as the discrimination between the data samples from different classes must be enhanced [14]. Then, through the same determined mapping/projection function $\mathbf{V}$, the corresponding test set $\widetilde{\mathbf{X}}$ are projected to the same embedded feature space, denoted by a $m \times k$ matrix $\widetilde{\mathbf{Z}}$, as:

$$\widetilde{\mathbf{Z}} = \widetilde{\mathbf{X}}\mathbf{V} = [\widetilde{\mathbf{z}}_1, \widetilde{\mathbf{z}}_2, \ldots, \widetilde{\mathbf{z}}_m]^T \tag{2.6}$$

where $\widetilde{\mathbf{z}}_i = \mathbf{V}^T\widetilde{\mathbf{x}}_i = [\widetilde{z}_{i1}, \widetilde{z}_{i2}, \ldots, \widetilde{z}_{ik}]^T$ corresponds to the ith embedded test data sample. The dimensionality of the data samples in the test set is reduced and the discrimination between the data samples from different classes should be enhanced. The

discriminability of the embedded test data samples is a way to measure the final performance of the determined mapping/projection function.

To give a better review of the existing embedding learning techniques, the well known techniques on unsupervised, supervised and semi-supervised learning are each examined in detail. Although the concept of multi-label learning is beyond the scope of this thesis, some commonly used techniques for multi-label classification are listed for completeness. Sometimes, since the meaningful structure of a data set lies on a nonlinear space, the linear embedding learning techniques will fail to discover the nonlinear structure. To overcome this problem, some techniques are developed to increase the power of the proposed algorithms for embedding learning by extending them to nonlinear space. Some commonly used techniques for nonlinear extension are also listed.

2.2 Unsupervised Learning Techniques

The unsupervised learning techniques determine mapping/projection functions on the basis of using some feature properties and characteristics of a set of data samples. Through the mapping/projection functions that are determined, the algorithms aim to provide a good and informative representation of the original set.

Principal Component Analysis

Principal Component Analysis (PCA) [42], also known as the Karhunen-Loeve Transform (KLT) [88], is one of the most widely used methods for linear embedding learning. PCA constructs a $d \times k$ orthogonal projection matrix $\mathbf{V}$, to maximise the variance of all the embedded data samples [42]. It has a number of advantages [89]: reducing the number of dimensions without much loss of the original information, removal of redundancy, and reduction of noise. For the training set $\mathbf{X}$, the maximisation of the variance of all the embeddings can be formulated as:

$$\arg\max_{\mathbf{z}_i \in R^k} \frac{1}{n-1} \sum_{i=1}^{n} \left\| \mathbf{z}_i - \frac{1}{n} \sum_{j=1}^{n} \mathbf{z}_j \right\|_2^2 \tag{2.7}$$

where $\| \cdot \|_2$ donates the 2-norm or Euclidean norm, and the objective function can be reformulated as [55]:

$$\begin{aligned} \sum_{i=1}^{n} \left\| \mathbf{z}_i - \frac{1}{n} \sum_{j=1}^{n} \mathbf{z}_j \right\|_2^2 &= \sum_{i=1}^{n} \left(\mathbf{z_i} - \frac{1}{n} \sum_{j=1}^{n} \mathbf{z_j} \right)^T \left(\mathbf{z_i} - \frac{1}{n} \sum_{j=1}^{n} \mathbf{z_j} \right) \\ &= \mathrm{tr} \left[\sum_{i=1}^{n} \left(\mathbf{z_i} - \frac{1}{n} \sum_{j=1}^{n} \mathbf{z_j} \right) \left(\mathbf{z_i} - \frac{1}{n} \sum_{j=1}^{n} \mathbf{z_j} \right)^T \right] \end{aligned} \tag{2.8}$$

Then, the maximisation problem in (eq. (2.7)) is equivalent to the following:

$$\underset{\mathbf{V}\in R^{d\times k},\mathbf{V}^T\mathbf{V}=\mathbf{I}_{k\times k}}{\arg\max} \operatorname{tr}\left[\sum_{i=1}^{n}\left(\mathbf{V}^T\mathbf{x}_i-\frac{1}{n}\sum_{j=1}^{n}\mathbf{V}^T\mathbf{x}_j\right)\left(\mathbf{V}^T\mathbf{x}_i-\frac{1}{n}\sum_{j=1}^{n}\mathbf{V}^T\mathbf{x}_j\right)^T\right] \tag{2.9}$$

where tr[·] is the trace operator, and the optimisation problem above can be reformulated as:

$$\underset{\mathbf{V}\in R^{d\times k},\mathbf{V}^T\mathbf{V}=\mathbf{I}_{k\times k}}{\arg\max} \operatorname{tr}\left[\mathbf{V}^T\mathbf{S}_t\mathbf{V}\right] \tag{2.10}$$

where $\mathbf{S}_t$ is defined as:

$$\mathbf{S}_t=\sum_{i=1}^{n}\left(\mathbf{x_i}-\frac{1}{n}\sum_{j=1}^{n}\mathbf{x_j}\right)\left(\mathbf{x_i}-\frac{1}{n}\sum_{j=1}^{n}\mathbf{x_j}\right)^T \tag{2.11}$$

$\mathbf{S}_t$ is called the total scatter matrix of the training set $\mathbf{X}$ [55]. $\mathbf{S}_t$ above can be expressed more succinctly, as:

$$\begin{aligned}\mathbf{S}_t&=\mathbf{X}^T\left(\mathbf{I}_{n\times n}-\frac{1}{n}\mathbf{1}_{n\times n}\right)^T\left(\mathbf{I}_{n\times n}-\frac{1}{n}\mathbf{1}_{n\times n}\right)\mathbf{X}\\&=\mathbf{X}^T\left(\mathbf{I}_{n\times n}-\frac{1}{n}\mathbf{1}_{n\times n}\right)\mathbf{X}\end{aligned} \tag{2.12}$$

where $\mathbf{I}_{n\times n}-\frac{1}{n}\mathbf{1}_{n\times n}$ is the $n\times n$ centring matrix.

The optimal projection matrix $\mathbf{V}^*=[\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_k]$ can be obtained by employing Lagrangian multipliers λ_i $(i=1,\ldots,k)$ [55]:

$$L(\mathbf{v}_i,\lambda_i)=\mathbf{v}_i^T\mathbf{S}_t\mathbf{v}_i-\lambda_i(\mathbf{v}_i^T\mathbf{v}_i-1) \tag{2.13}$$

Next the derivative with respect to $\mathbf{v}_i$ is taken:

$$\frac{\partial L(\mathbf{v}_i,\lambda_i)}{\partial\mathbf{v}_i}=\mathbf{S}_t\mathbf{v}_i+\mathbf{S}_t^T\mathbf{v}_i-2\lambda_i\mathbf{v}_i \tag{2.14}$$

using the fact that $\mathbf{S}_t$ is a symmetric matrix, and the derivative result is set to zero, thus the following is obtained:

$$\mathbf{S}_t\mathbf{v}_i=\lambda_i\mathbf{v}_i \tag{2.15}$$

Clearly, the equation above is a standard eigenvalue problem. The optimal projection matrix $\mathbf{V}^*$ is obtained by Singular Value Decomposition (SVD) [90], whose columns $[\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_k]$ are the k standard eigenvectors of $\mathbf{S}_t$, corresponding to the k largest standard eigenvalues $[\lambda_1,\lambda_2,\ldots,\lambda_k]$.

An example of PCA embedding learning for a set of data samples in 2D is shown in Figure 2.1. The data samples are generated from a multivariate normal distribution with a mean of $(0.392, 0.207)$ and a covariance of $[1, 1.5; 1.5, 3]$, and then the PCA feature space is found (i.e. the red line shown in Figure 2.1a) so that when all the

data samples are projected from the original 2D feature space into this new 1D feature space, the corresponding variance is maximised.

To visually analyse the projected data samples in the determined 1D feature space, the corresponding density can be estimated as follows: First, let m_{line} represent the slope or gradient of the determined line, then the projected coordinate of an arbitrary data point x_i can be calculated as:

$$y_i = [\cos(\arctan(m_{\text{line}})), \sin(\arctan(m_{\text{line}}))] \times [x_{i1}, x_{i2}]^T \tag{2.16}$$

Since a linear shift in the projected coordinate set $\mathbf{Y}$ will not have any effect on the final result of density estimation, $\mathbf{Y}$ can be shifted linearly so that it starts from the value of 0. This can be simply achieved by:

$$\mathbf{Y} = \mathbf{Y} - \min(\mathbf{Y}) \tag{2.17}$$

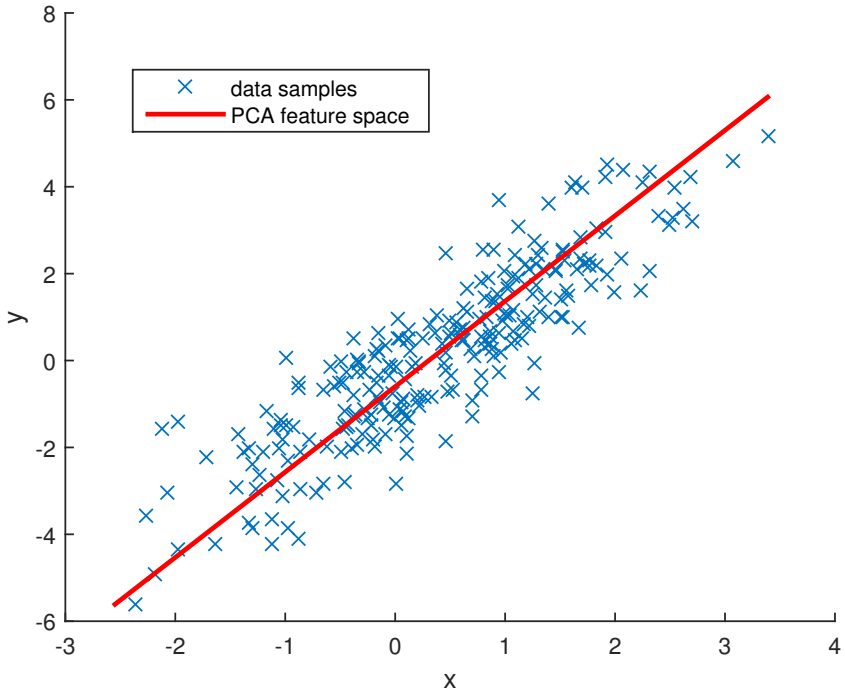
where $\min(\cdot)$ returns the minimum value. Next, a histogram, with the number of bins being 20, is contracted based on the shifted coordinate set $\mathbf{Y}$. Finally, a density function is estimated by fitting it to the contracted histogram, based on the assumption that the data samples are in a nonparametric kernel-smoothing distribution.

Latent Semantic Indexing Latent Semantic Indexing (LSI) [43] is a method that works very similarly to PCA. It operates like PCA, but without the process of centring the data samples eq. (2.12). A document-term matrix, which identifies the occurrences of a number of unique terms within a dataset generated by a collection of document-like data (e.g. 1 indicates a term occurs, while 0 otherwise), is usually a sparse matrix. Thus, the centring process (i.e. subtracting off the mean) will result in losing the original sparseness present in the document-term matrix, and then increasing the cost of processing and/or storing it. LSI was developed for text processing, however due to its similarity to PCA, it has been applied to a variety of application areas, such as information retrieval. Based on (2.10) and (2.12), LSI gets its optimal projection function by solving the following optimisation problem:

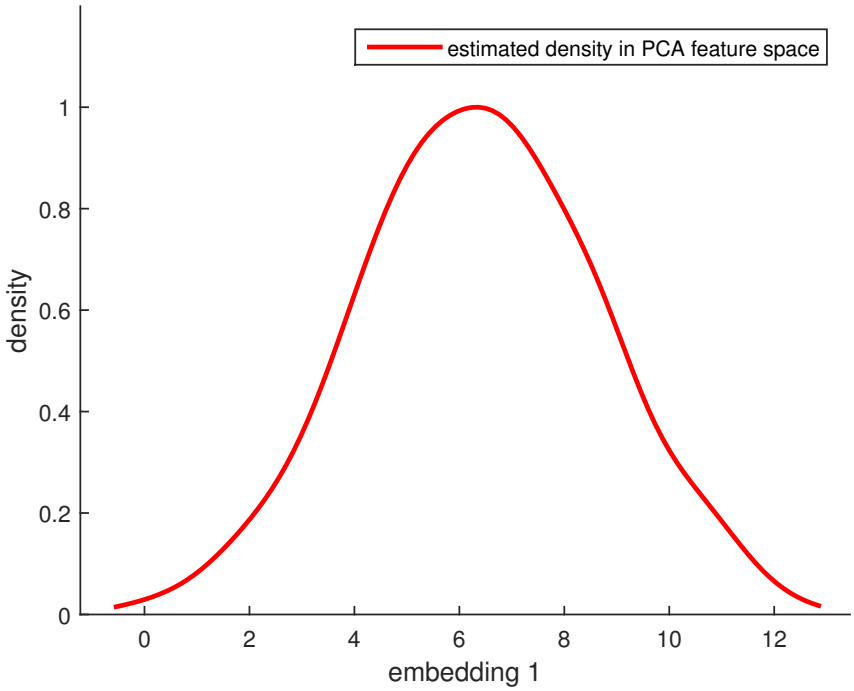
$$\underset{\mathbf{V} \in R^{d \times k}, \mathbf{V}^T\mathbf{V} = \mathbf{I}_{k \times k}}{\arg\max} \operatorname{tr}\left[\mathbf{V}^T\mathbf{X}^T\mathbf{X}\mathbf{V}\right] \tag{2.18}$$

Locality Preserving Projection

Both PCA and LSI construct the optimal mapping functions based on preserving the global structure of the original set, because of this it is likely that the corresponding local structures are distorted. Locality Preserving Projection (LPP) [44] is a method that was proposed to focus more on the local structures of the original set, when constructing the optimal mapping function. It is developed through the pairwise proximity information, which is based on a constructed graph of local neighbours that is used to



(a)



(b) the estimated density of the projected data samples in the PCA feature space

Figure 2.1: An example of embedding learning - PCA: (a) for a set of 2D data samples generated from a multivariate normal distribution, a PCA feature space is calculated so that the corresponding variance is maximised.

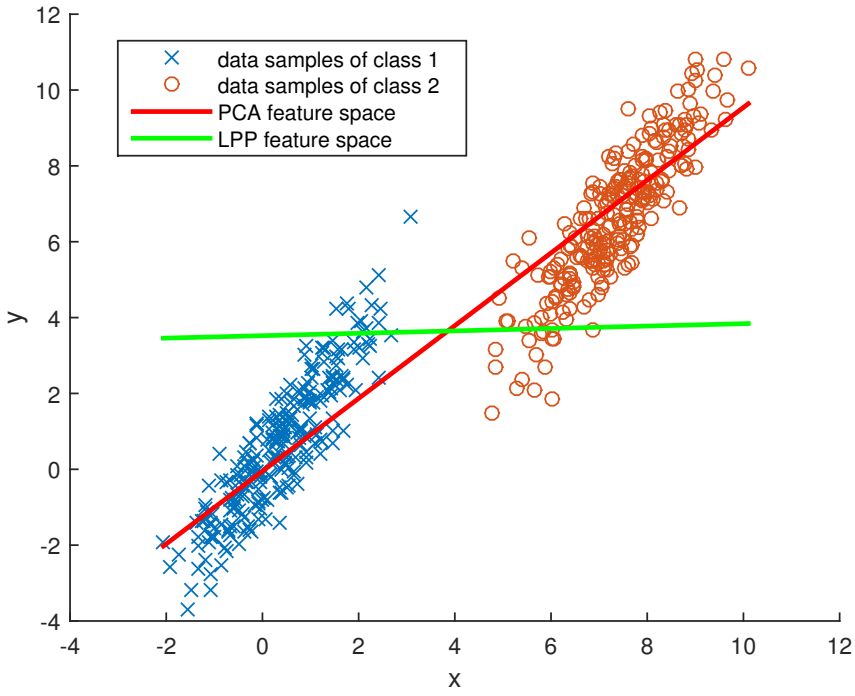
represent the local structures. As a result, it is likely that a search of nearest neighbours in the resultant embedding feature space will yield a similar result, compared to that in the original high dimensional feature space. This makes LPP particularly useful in applications such as information retrieval, where a search of a nearest neighbour is ultimately needed [44]. LPP has been demonstrated to have more discriminating power than PCA and be insensitive to the presence of outliers [44]. The advantages are both shown as an example of embedding learning between PCA and LPP in Figure 2.2 and 2.3. In Figure 2.2a, the data samples of two different classes are generated from two different multivariate normal distributions, with different means $(0.366, 0.620)$ and $(7.366, 6.620)$ while a common covariance $[1, 1.5; 1.5, 3]$. Comparing Figure 2.2b and 2.2c, it can be seen that the two estimated densities shown in Figure 2.2c have less overlap region. It means that in the LPP feature space, the projected data samples have better separation performance. Because a suitable distance is maintained between the data samples of the two classes, a simple classifier based on nearest neighbour search (e.g. 1-NN, which simply assigns a test data sample to the class of its nearest neighbour) is powerful enough to achieve class discrimination. PCA feature space, although it maximises the variance, it distorts the good arrangement between nearest neighbour and class label (i.e. a data sample and its nearest neighbour are from the same class). This makes the 1-NN classifier fail to achieve a good performance. The LPP feature space preserves the local structures, which results in more discriminating power. In Figure 2.3a, the data samples are generated from a multivariate normal distribution with a mean of $(0.879, 0.548)$ and a covariance of $[1, 1.5; 1.5, 3]$, while several outliers are added to the Gaussians. For the same reason, the resultant LPP feature space is less sensitive to outliers, compared to the PCA one. However, LPP may lead to unsatisfying effectiveness due to the constructed graph of local neighbours [41]. This is because the local neighbourhood graph is generated based on adopting the Euclidean distance as the similarity measure between data samples, while the Euclidean distance cannot adequately describe the intrinsic geometry of the manifold structure in the real word [41].

LPP is a linear approximation of Laplacian Eigenmaps (LE) [34], where a linear constraint is imposed between the original set and the determined mapping/projection function. It attempts to minimise the pairwise distance errors between all the data samples in the embedded feature space, and then for the training set $\mathbf{X}$, the corresponding minimisation problem is defined as:

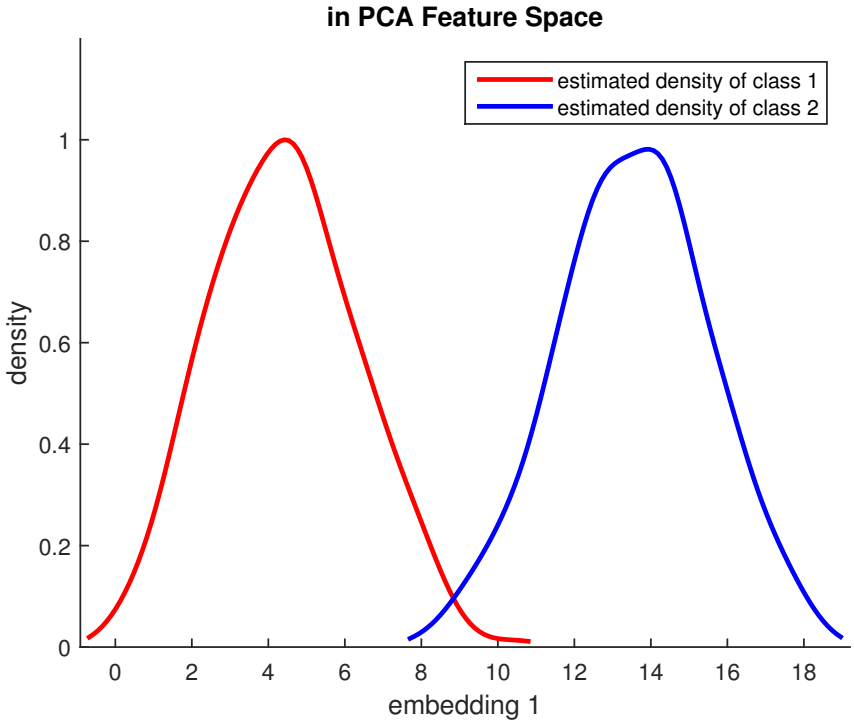
$$\underset{\mathbf{z}_i \in R^k}{\arg\min} \sum_{i,j=1}^{n} w_{ij} \left\| \mathbf{z}_i - \mathbf{z}_j \right\|_2^2 \tag{2.19}$$

where the imposed linear constraint is $\mathbf{z}_i = \mathbf{V}^T \mathbf{x}_i$, and w_{ij} is the ijth element of the $n \times n$ weight matrix $\mathbf{W}$. w_{ij} is defined based on a similarity or closeness measure between the ith and jth data samples in the original feature space. The commonly

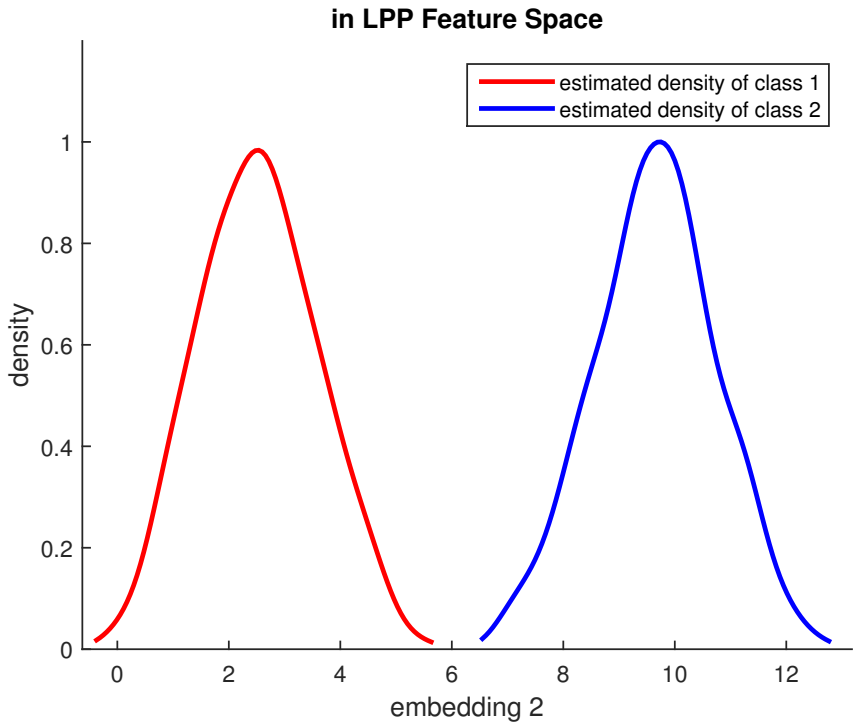
used similarity or closeness measures for weight matrix $\mathbf{W}$ are listed in appendix A. w_{ij} is only nonzero when the ith and jth data samples are connected. Commonly, two different ways are used to define whether a connection exists. One is when the ith data



(a)



(b) the estimated densities of the projected data samples in the PCA feature space



(c) the estimated densities of the projected data samples in the LPP feature space

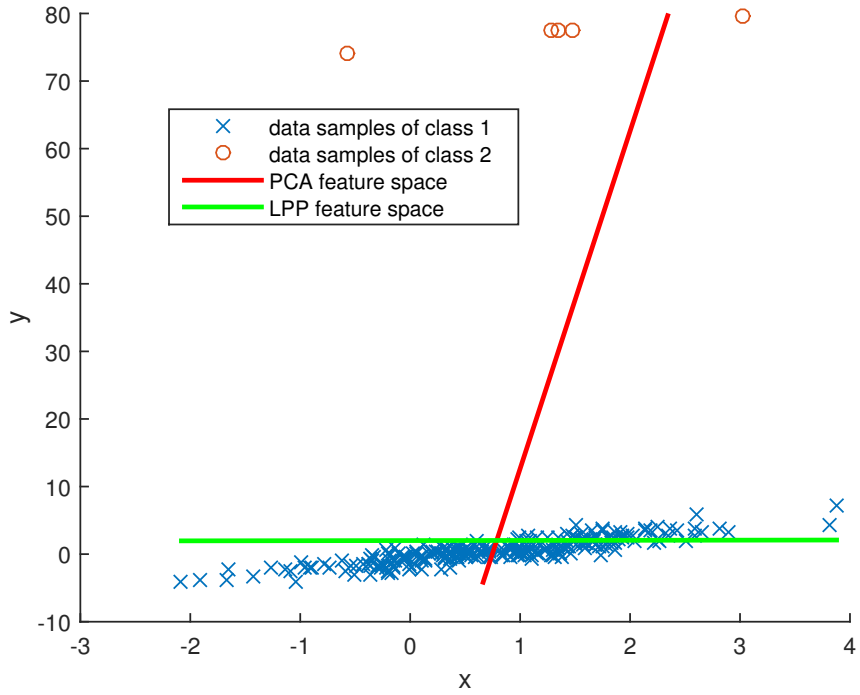
Figure 2.2: An example of embedding learning - LPP versus PCA: (a) the projections of the original 2D data in the LPP feature space have more discriminating power, compared to those in the PCA feature space

sample is the mutual or undirected k-NN of the jth one, and vice versa; the other is when a specific measure of the distance between the ith and jth data samples is below a pre-defined threshold. Based on the work in [44], the objective function above can be reduced to:

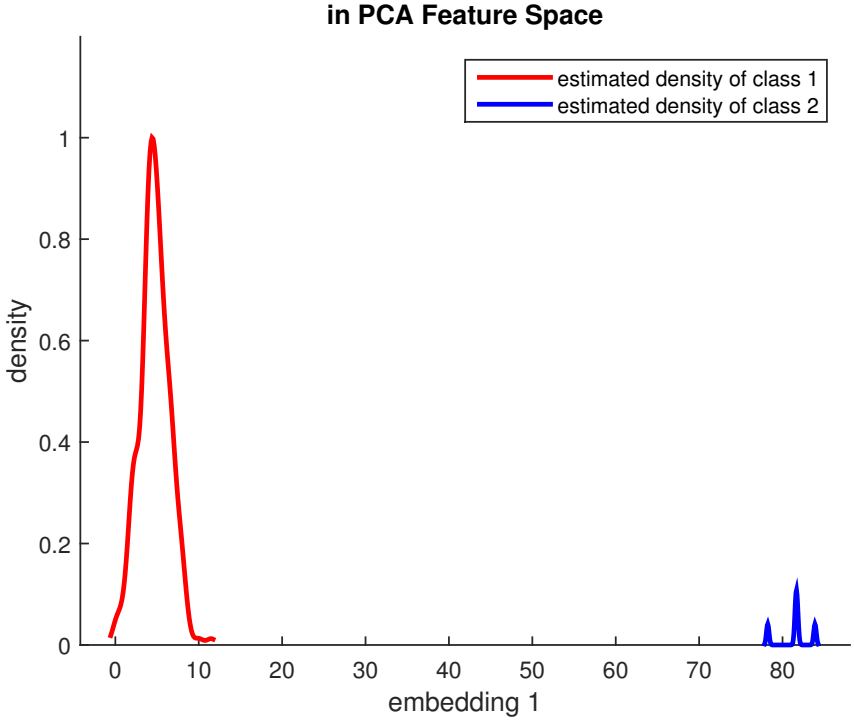
$$\begin{aligned}
\frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\left\|\mathbf{z}_i-\mathbf{z}_j\right\|_2^2 &= \frac{1}{2}\left(\sum_{i,j=1}^{n} w_{ij}\mathbf{z}_i^T\mathbf{z}_i+\sum_{i,j=1}^{n} w_{ij}\mathbf{z}_j^T\mathbf{z}_j-\sum_{i,j=1}^{n} w_{ij}\mathbf{z}_i^T\mathbf{z}_j-\sum_{i,j=1}^{n} w_{ij}\mathbf{z}_j^T\mathbf{z}_i\right)\\
&= \sum_{i,j=1}^{n} w_{ij}\mathbf{z}_i^T\mathbf{z}_i-\sum_{i,j=1}^{n} w_{ij}\mathbf{z}_i^T\mathbf{z}_j\\
&= \sum_{i=1}^{n}\mathbf{z}_i^T\left(\sum_{j=1}^{n} w_{ij}\right)\mathbf{z}_i-\sum_{i,j=1}^{n}\mathbf{z}_i^T w_{ij}\mathbf{z}_j\\
&= \mathrm{tr}\left[\sum_{i=1}^{n}\mathbf{z}_i\left(\sum_{j=1}^{n} w_{ij}\right)\mathbf{z}_i^T-\sum_{i,j=1}^{n}\mathbf{z}_i w_{ij}\mathbf{z}_j^T\right]\\
&= \mathrm{tr}\left[\left(\mathbf{Z}^T\mathbf{D}\mathbf{Z}-\mathbf{Z}^T\mathbf{W}\mathbf{Z}\right)\right] \qquad (2.20)
\end{aligned}$$

where $\mathbf{D}$ is an $n \times n$ diagonal matrix, with its ith diagonal element d_{ii} being:

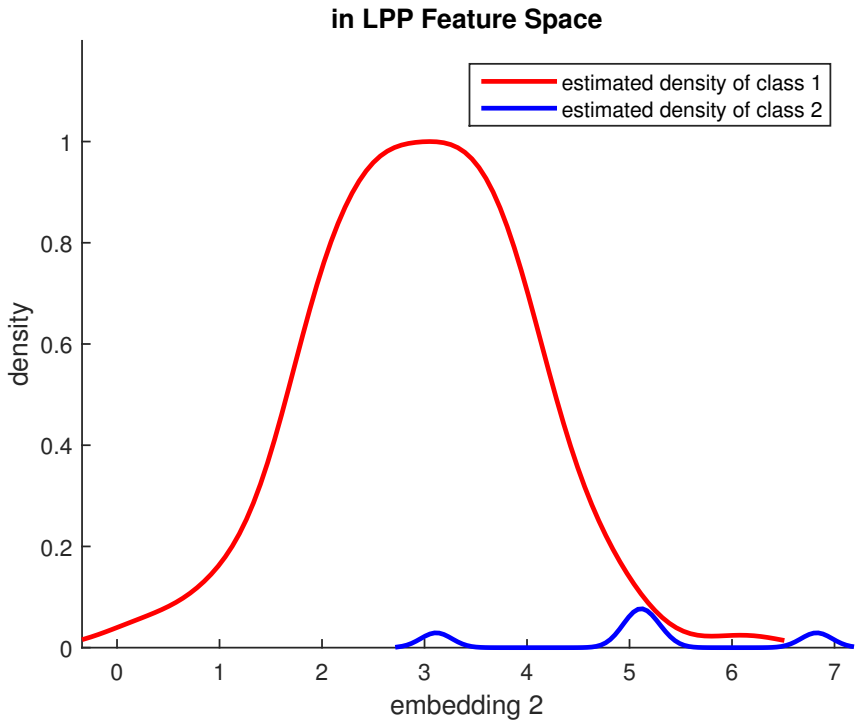
$$d_{ii} = \sum_{j=1}^{n} w_{ij} \tag{2.21}$$



(a)



(b) the estimated densities of the projected data samples in the PCA feature space



(c) the estimated densities of the projected data samples in the LPP feature space

Figure 2.3: An example of embedding learning - LPP versus PCA (taken from [44] but redrawn here): (a) compared to the feature space determined by PCA, the feature space determined by LPP is less sensitive to the existing outliers.

and it can be formed from [15]:

$$\mathbf{D} = \text{diag}(\mathbf{W} \times \mathbf{1}_{n\times 1}) \tag{2.22}$$

where $\mathbf{D}$ is a diagonal matrix, the function diag$(\cdot)$ returns a vector corresponding to the matrix diagonal for a given input matrix or a corresponding diagonal matrix for an input vector, and $\mathbf{1}_{n\times 1}$ is a n-dimensional vector with all its elements to be 1. Thus, the minimisation problem in (2.19) can be reduced to finding the following:

$$\underset{\mathbf{V}\in R^{d\times k}, \mathbf{V}^T\mathbf{X}^T\mathbf{D}\mathbf{X}\mathbf{V}=\mathbf{I}_{k\times k}}{\arg\min} \text{tr}\left[\mathbf{V}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{V}\right] \tag{2.23}$$

where the constraint $\mathbf{V}^T\mathbf{X}^T\mathbf{D}\mathbf{X}\mathbf{V} = \mathbf{I}_{k\times k}$ is imposed in [44] to remove arbitrary scaling factors in the embeddings, and $\mathbf{L}$ is an $n \times n$ Laplacian matrix, which is defined as:

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{2.24}$$

Similarly, introducing Lagrangian multipliers λ_i $(i = 1, \ldots, k)$, the solution to the minimisation problem in (2.23) can be reduced to the following generalised eigenvalue problem:

$$\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{v}_i = \lambda_i\mathbf{X}^T\mathbf{D}\mathbf{X}\mathbf{v}_i \tag{2.25}$$

The optimal projection matrix $\mathbf{V}^*$ is then obtained, whose columns $[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$ are the k generalised eigenvectors of $\mathbf{X}^T\mathbf{L}\mathbf{X}$ and $\mathbf{X}^T\mathbf{D}\mathbf{X}$, corresponding to the k smallest generalised eigenvalues $[\lambda_1, \lambda_2, \ldots, \lambda_k]$.

Orthogonal Locality Preserving Projection Orthogonal Locality Preserving Projection (OLPP) [45] is different from LPP by modifying the constraint to be $\mathbf{V}^T\mathbf{V} = \mathbf{I}_{k\times k}$, to make the columns of the projection $\mathbf{V}$ be orthonormal. The metric structure in the resultant embedded feature space can be well persevered, because the projection function has orthogonal columns [91]. Additionally, OLPP preserves the global geometry, as the orthogonal projection function is faithful, where it carries information about the fold structure of the manifold in the original high dimensional feature space. [45]. Following (2.23) OLPP imposes its optimal projection function by solving the following optimisation problem:

$$\underset{\mathbf{V}\in R^{d\times k}, \mathbf{V}^T\mathbf{V}=\mathbf{I}_{k\times k}}{\arg\min} \operatorname{tr}\left[\mathbf{V}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{V}\right] \tag{2.26}$$

Orthogonal Neighbourhood Preserving Projection

Orthogonal Neighbourhood Preserving Projections (ONPP) [45] is an another popular method that considers local structures. Different from LPP, it is a linear variation of Locally Linear Embedding (LLE) [13, 92], where it assumes that each data sample, along with its k-NNs (approximately) lie on a locally linear manifold. ONPP aims to preserve the intrinsic geometry of the local neighbours, whereas LPP only aims to preserve locality without explicit considering the structures of the local geometry [45]. ONPP attempts to minimise the differences between each embedded data sample and its reconstruction that are formed by a linear combination of its k-NNs, as:

$$\underset{\mathbf{z}_i\in R^d}{\arg\min} \sum_{i=1}^{n} \left\| \mathbf{z}_i - \sum_{j=1}^{n} w_{ij}^{\text{lle}} \mathbf{z}_j \right\|_2^2 \tag{2.27}$$

where w_{ij}^{lle} is the ij element of the pairwise weight matrix $\mathbf{W}^{\text{lle}}$. Different from the pairwise weight defined in LPP, the w_{ij}^{lle} used here are the same as that used in LLE. It is a linear coefficient based on the reconstruction from the local neighbours. The way to compute the LLE style weight is given in appendix A.1. w_{ij} is only nonzero, if the data sample $\mathbf{x}_j$ is one of the k-NN of $\mathbf{x}_i$. Additionally, the coefficients are rescaled so that $\sum_{j=1}^{n} w_{ij}^{\text{lle}} = 1$ to make the reconstruction of data sample $\mathbf{x}_i$ a convex combination of its k-NN [45]. The objective function above can be reduced to give:

$$\underset{\mathbf{V}\in R^{d\times k}, \mathbf{V}^T\mathbf{V}=\mathbf{I}_{k\times k}}{\arg\min} \operatorname{tr}\left[\mathbf{V}^T \left(\sum_{i=1}^{n} \left(\mathbf{x}_i - \sum_{j=1}^{n} w_{ij}^{\text{lle}} \mathbf{x}_j \right) \left(\mathbf{x}_i - \sum_{j=1}^{n} w_{ij}^{\text{lle}} \mathbf{x}_j \right)^T \right) \mathbf{V}\right] \tag{2.28}$$

It can be seen that, the above objective function shares the same structure as the one in (2.9) and (2.12), where in PCA, w_{ij}^{lle} is set to be the constant value $1/n$. Thus,

$$\sum_{i=1}^{n}\left(\mathbf{x}_i - \sum_{j=1}^{n} w_{ij}^{\text{lle}}\mathbf{x}_j\right) = (\mathbf{I}_{n\times n} - \mathbf{W})\,\mathbf{X} \tag{2.29}$$

The maximisation problem defined in (2.27) can be reformulated as [45]:

$$\underset{\mathbf{V}\in R^{d\times k},\mathbf{V}^T\mathbf{V}=\mathbf{I}_{k\times k}}{\arg\min}\ \text{tr}\left[\mathbf{V}^T\mathbf{X}^T\left(\mathbf{I}_{n\times n} - \mathbf{W}^{\text{lle}}\right)^T\left(\mathbf{I}_{n\times n} - \mathbf{W}^{\text{lle}}\right)\mathbf{X}\mathbf{V}\right] \tag{2.30}$$

Similar to PCA, the optimal projection matrix $\mathbf{V}^*$ can then be obtained by employing Lagrange multipliers. Its columns $[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$ are the k standard eigenvectors of $\mathbf{X}^T(\mathbf{I}_{n\times n} - \mathbf{W}^{\text{lle}})^T(\mathbf{I}_{n\times n} - \mathbf{W}^{\text{lle}})\mathbf{X}$, corresponding to the k smallest standard eigenvalues $[\lambda_1, \lambda_2, \ldots, \lambda_k]$.

2.3 Supervised Learning Techniques

LPP, ONPP and various other methods of manifold learning and spectral analysis can achieve a high final performance when there exists a good arrangement between the features and the labels in a dataset (i.e. in the feature space, the data samples in the same class are located nearby, while those from different classes are located far away). But in real time applications, this does not often hold, which makes methods like LPP lead to increased final classification errors. In such cases, supervised learning techniques, which utilise label information to construct the mapping/projection functions, tend to be preferred.

Fisher Discriminant Analysis

Fisher Discriminant Analysis (FDA) is a commonly used supervised method [50]. It computes the mapping function based on maximising the variance of the embeddings between different classes, while simultaneously minimising that in the same class. For the training set $\mathbf{X}$, it is defined as [55]:

$$\underset{\mathbf{z}_i\in R^k}{\arg\max}\ \frac{\sum_{t=1}^{c} n_t \left\| \frac{1}{n_t}\sum_{\mathbf{z}_i\in C_t}\mathbf{z}_i - \frac{1}{n}\sum_{j=1}^{n}\mathbf{z}_j \right\|_2^2}{\sum_{t=1}^{c}\sum_{\mathbf{z}_i\in C_t}\left\| \mathbf{z}_i - \frac{1}{n_t}\sum_{\mathbf{z}_j\in C_t}\mathbf{z}_j \right\|_2^2} \tag{2.31}$$

where C_t is the tth class. The above maximisation problem can be reformulated to be the following [55]:

$$\underset{\mathbf{V}\in R^{d\times k}}{\arg\max} \frac{\operatorname{tr}\left[\sum_{t=1}^{c} n_t \left(\frac{1}{n_t}\sum_{\mathbf{x}_i\in C_t}\mathbf{V}^T\mathbf{x}_i - \frac{1}{n}\sum_{j=1}^{n}\mathbf{V}^T\mathbf{x}_j\right)\left(\frac{1}{n_t}\sum_{\mathbf{x}_i\in C_t}\mathbf{V}^T\mathbf{x}_i - \frac{1}{n}\sum_{j=1}^{n}\mathbf{V}^T\mathbf{x}_j\right)^T\right]}{\operatorname{tr}\left[\sum_{t=1}^{c}\sum_{\mathbf{x}_i\in C_t}\left(\mathbf{V}^T\mathbf{x}_i - \frac{1}{n_t}\sum_{\mathbf{x}_j\in C_t}\mathbf{V}^T\mathbf{x}_j\right)\left(\mathbf{V}^T\mathbf{x}_i - \frac{1}{n_t}\sum_{\mathbf{x}_j\in C_t}\mathbf{V}^T\mathbf{x}_j\right)^T\right]} \tag{2.32}$$

This optimisation problem can be reduced to give [55]:

$$\underset{\mathbf{V}\in R^{d\times k}}{\arg\max} \frac{\operatorname{tr}\left[\mathbf{V}^T\mathbf{S}_b\mathbf{V}\right]}{\operatorname{tr}\left[\mathbf{V}^T\mathbf{S}_w\mathbf{V}\right]} \tag{2.33}$$

where $\mathbf{S}_b$ and $\mathbf{S}_w$ are the between-class and within-class scatter matrices respectively, and they are calculated using:

$$\mathbf{S}_b = \sum_{t=1}^{c} n_t \left(\frac{1}{n_t}\sum_{\mathbf{x}_i\in C_t}\mathbf{x}_i - \frac{1}{n}\sum_{j=1}^{n}\mathbf{x}_j\right)\left(\frac{1}{n_t}\sum_{\mathbf{x}_i\in C_t}\mathbf{x}_i - \frac{1}{n}\sum_{j=1}^{n}\mathbf{x}_j\right)^T \tag{2.34}$$

and

$$\mathbf{S}_w = \sum_{t=1}^{c}\sum_{\mathbf{x}_i\in C_t}\left(\mathbf{x}_i - \frac{1}{n_t}\sum_{\mathbf{x}_j\in C_t}\mathbf{x}_j\right)\left(\mathbf{x}_i - \frac{1}{n_t}\sum_{\mathbf{x}_j\in C_t}\mathbf{x}_j\right)^T \tag{2.35}$$

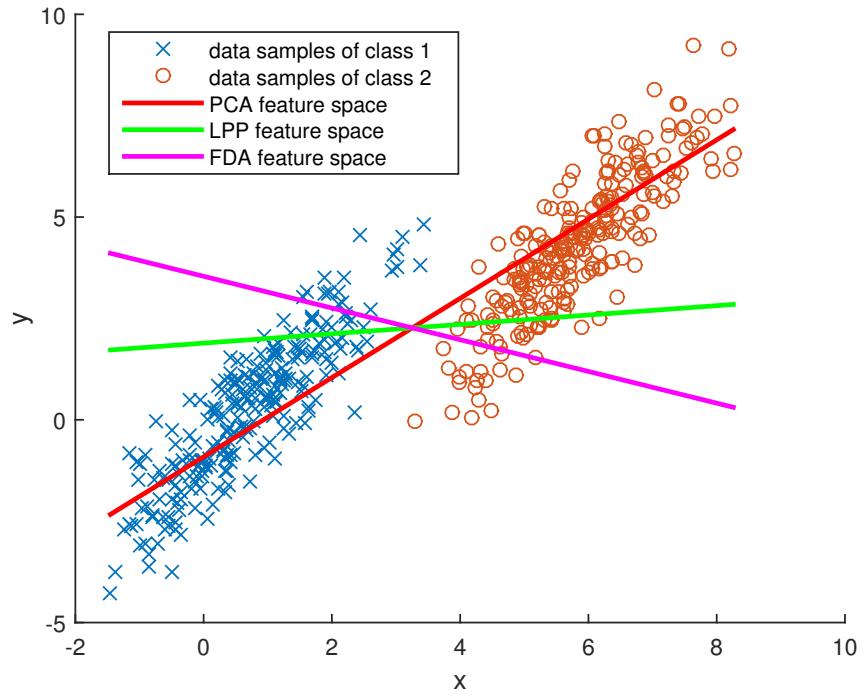
Since $\mathbf{V}^T\mathbf{S}_w\mathbf{V}$ is a scalar measure, $\mathbf{V}$ can always be chosen such that $\mathbf{V}^T\mathbf{S}_w\mathbf{V} = I_{k\times k}$. Thus, the maximisation problem above can be transformed into the following constrained optimisation problem:

$$\underset{\mathbf{V}\in R^{d\times k}, \mathbf{V}^T\mathbf{S}_w\mathbf{V}=I_{k\times k}}{\arg\max} \operatorname{tr}\left[\mathbf{V}^T\mathbf{S}_b\mathbf{V}\right] \tag{2.36}$$

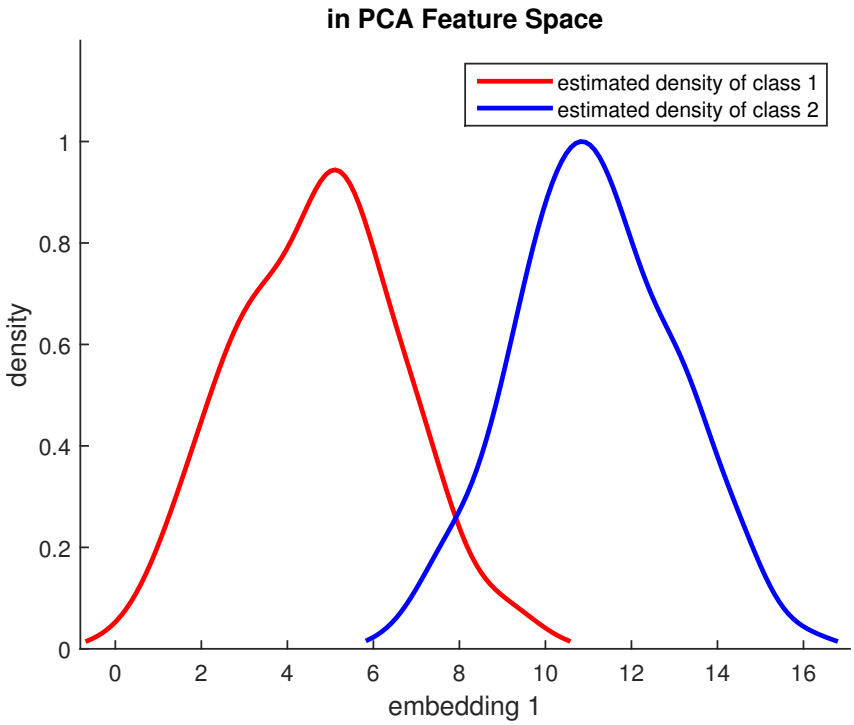
As with LPP, the solution (i.e. the optimal projection matrix $\mathbf{V}^*$) can be obtained with its columns $[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$ being the k generalised eigenvectors of $\mathbf{S}_b$ and $\mathbf{S}_w$, corresponding to the k largest generalised eigenvalues $[\lambda_1, \lambda_2, \ldots, \lambda_k]$.

An example of embedding learning that compares the learning results of FDA, LPP and PCA is shown in Figure 2.4a. The data samples of two different classes are generated from two multivariate normal distributions (with different means $(0.725, 0.248)$ and $(5.725, 4.248)$ while a common covariance $[1, 1.5; 1.5, 3]$), where there is not a good arrangement between data samples and class labels. In other words, the nearest neighbours of some data samples include ones from the other class. As a result, the FDA feature space that is determined based on the label information shows the most discriminant ability, compared to the PCA and LPP feature spaces (i.e. in the FDA feature space, the projections of the data samples of intraclass stay close together, while those of interclass maintain a suitable distance).

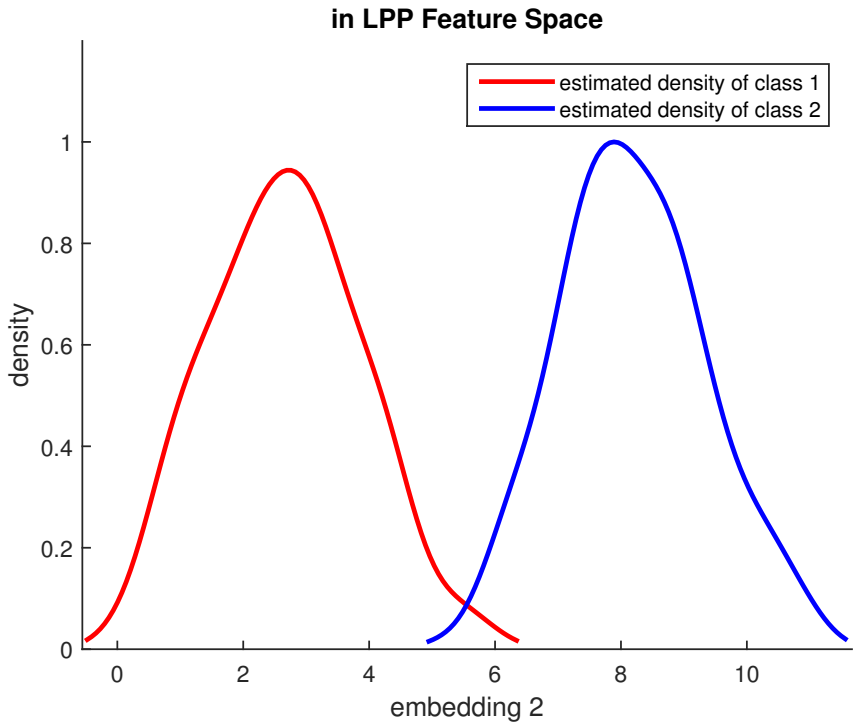
Maximum Margin Criterion Maximum Margin Criterion (MMC) [51] was proposed to achieve the same goal as FDA. It utilises the criterion that takes the difference between the between class scatter matrix $\mathbf{S}_b$ and the within class scatter matrix $\mathbf{S}_w$,



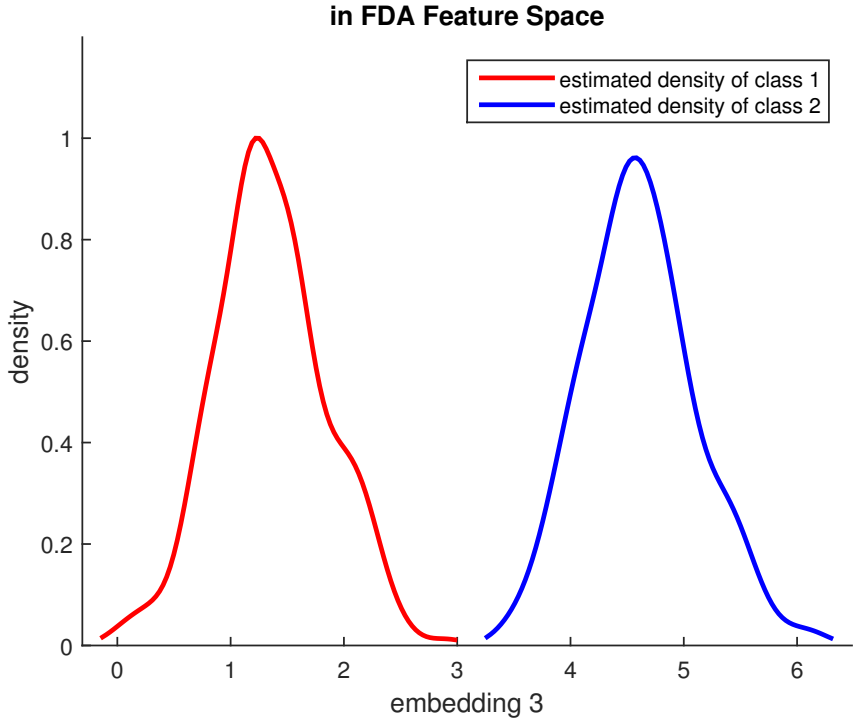
(a)



(b) the estimated densities of the projected data samples in the PCA feature space



(c) the estimated densities of the projected data samples in the LPP feature space



(d) the estimated densities of the projected data samples in the FDA feature space

Figure 2.4: An example of embedding learning - FDA versus LPP versus PCA: (a) the projections of the original 2D data in the FDA feature space possess the most discriminant ability.

rather than the ratio of the two. Taking the difference instead of the ratio results in avoiding the need to calculate the inverse of $\mathbf{S}_w$, and the potential small sample size problem [51]. The small sample size problem [93] refers to the situation where the number of available data samples is much smaller than the corresponding dimensionality. It is often encountered in real-time applications (e.g. face recognition) [51], and it tends to lead to a high probability of Type II error, which corresponds to a low statistical power [94, 95]. Here, it will result in the within class scatter matrix $\mathbf{S}_w$ being singular. Compared to FDA, MMC additionally employs the constraint $\mathbf{V}^T\mathbf{V} = \mathbf{I}_{n\times n}$ to ensure the resultant projection function is orthonormal. It determines the optimal mapping function using the following maximisation problem:

$$\underset{\mathbf{V}\in R^{d\times k}, \mathbf{V}^T\mathbf{V}=\mathbf{I}_{k\times k}}{\arg\max} \operatorname{tr}\left[\mathbf{V}^T(\mathbf{S}_b - \mathbf{S}_w)\mathbf{V}\right] \tag{2.37}$$

Regularised Maximum Margin Criterion Regularised Maximum Margin Criterion (rMMC) is another version of MMC proposed in [68]. It employs a regularisation parameter β to improve generalisation. It results in the original objection function of MMC being penalised, in order to prevent overfitting (i.e. it is more accurate when fitting data samples from the training set, but less accurate when predicting ones from the test set). It is defined as:

$$\underset{\mathbf{V}\in R^{d\times k}, \mathbf{V}^T\mathbf{V}=\mathbf{I}_{k\times k}}{\arg\max} \operatorname{tr}\left[\mathbf{V}^T(\mathbf{S}_b - \beta\mathbf{S}_w)\mathbf{V}\right] \tag{2.38}$$

where $\beta > 0$.

Another group of methods that are motivated by the same idea [52, 53, 56, 57], employ different types of metric to set up the between-class and within-class measures. For example, Local Fisher Discriminant Analysis (LFDA) [57] incorporates the idea of LPP into that of FDA, which considers the local information when defining the between-class and within-class scatter matrices.

Local Fisher Discriminant Analysis

As with PCA, FDA suffers the problem of distortion of local structures because it only takes the global structure into account. Local Fisher Discriminant Analysis (LFDA) [57] is one of the methods that are motivated by FDA, but are modified to incorporate the local information redefining the between-class and within-class scatters. Firstly, it

reformulates the within-class scatter matrix $\mathbf{S}_w$ defined in FDA to give:

$$\begin{aligned}
\mathbf{S}_w =& \sum_{t=1}^{c} \sum_{\mathbf{x}_i \in C_t} \mathbf{x}_i \mathbf{x}_i^T - \sum_{t=1}^{c} \sum_{\mathbf{x}_i \in C_t} \frac{1}{n_t} \sum_{\mathbf{x}_j \in C_t} \mathbf{x}_i \mathbf{x}_j^T - \sum_{t=1}^{c} \sum_{\mathbf{x}_i \in C_t} \frac{1}{n_t} \sum_{\mathbf{x}_j \in C_t} \mathbf{x}_j \mathbf{x}_i^T \\
&+ \sum_{t=1}^{c} \sum_{\mathbf{x}_i \in C_t} \frac{1}{n_t^2} \sum_{\mathbf{x}_j \in C_t} \sum_{\mathbf{x}_j \in C_t} \mathbf{x}_j \mathbf{x}_j^T \\
=& \sum_{t=1}^{c} \sum_{\mathbf{x}_i \in C_t} \mathbf{x}_i \mathbf{x}_i^T - \sum_{t=1}^{c} \sum_{\mathbf{x}_i \in C_t} \frac{1}{n_t} \sum_{\mathbf{x}_j \in C_t} \mathbf{x}_i \mathbf{x}_j^T \\
=& \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T - \sum_{t=1}^{c} \frac{1}{n_t} \sum_{\mathbf{x}_i, \mathbf{x}_j \in C_t} \mathbf{x}_i \mathbf{x}_j^T \\
=& \sum_{i=1}^{n} \left(\sum_{j=1}^{n} w_{ij}^w \right) \mathbf{x}_i \mathbf{x}_i^T - \sum_{i,j=1}^{n} w_{ij}^w \mathbf{x}_i \mathbf{x}_j^T \\
=& \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}^w \left(\mathbf{x}_i \mathbf{x}_i^T - \mathbf{x}_i \mathbf{x}_j^T - \mathbf{x}_j \mathbf{x}_i^T + \mathbf{x}_j \mathbf{x}_j^T \right) \\
=& \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}^w \left(\mathbf{x}_i - \mathbf{x}_j \right) \left(\mathbf{x}_i - \mathbf{x}_j \right)^T
\end{aligned} \tag{2.39}$$

where w_{ij}^w is the ijth element of the within-class weight matrix $\mathbf{W}^w$, and it is defined as:

$$w_{ij}^w = \begin{cases} 1/n_t & \text{if } \mathbf{x}_i,\, \mathbf{x}_j \in C_t \\ 0 & \text{otherwise} \end{cases} \tag{2.40}$$

From the fact that $\mathbf{S}_t = \mathbf{S}_w + \mathbf{S}_b$ ($\mathbf{S}_t$ is defined in (2.11)), the between-class scatter $\mathbf{S}_b$ can be obtained as:

$$\begin{aligned}
\mathbf{S}_b &= \mathbf{S}_t - \mathbf{S}_w \\
&= \sum_{i=1}^{n} \left(\mathbf{x}_i - \frac{1}{n} \sum_{j=1}^{n} \mathbf{x_j} \right) \left(\mathbf{x}_i - \frac{1}{n} \sum_{j=1}^{n} \mathbf{x}_j \right)^T - \mathbf{S}_w \\
&= \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T - \frac{1}{n} \sum_{i,j=1}^{n} \mathbf{x}_i \mathbf{x}_j - \mathbf{S}_w \\
&= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \frac{1}{n} \right) \mathbf{x}_i \mathbf{x}_i^T - \frac{1}{n} \sum_{i,j=1}^{n} \mathbf{x}_i \mathbf{x}_j - \mathbf{S}_w \\
&= \frac{1}{2} \sum_{i,j=1}^{n} \frac{1}{n} \left(\mathbf{x}_i \mathbf{x}_i^T - \mathbf{x}_i \mathbf{x}_j^T - \mathbf{x}_j \mathbf{x}_i^T + \mathbf{x}_j \mathbf{x}_j^T \right) - \mathbf{S}_w \\
&= \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}^b \left(\mathbf{x}_i - \mathbf{x}_j \right) \left(\mathbf{x}_i - \mathbf{x}_j \right)^T
\end{aligned} \tag{2.41}$$

where w_{ij}^b is the ijth element of the between-class weight matrix $\mathbf{W}^b$, and it is defined as:

$$w_{ij}^b = \begin{cases} 1/n - 1/n_t & \text{if } \mathbf{x}_i,\, \mathbf{x}_j \in C_t \\ 1/n & \text{otherwise} \end{cases} \tag{2.42}$$

Then, to take into account the local information, the local between-class scatter $\widetilde{\mathbf{S}}^b$ and local within-class scatter $\widetilde{\mathbf{S}}^w$ are defined to be:

$$\widetilde{\mathbf{S}}_b = \frac{1}{2}\sum_{i,j=1}^{n} \widetilde{w}_{ij}^b \left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \tag{2.43}$$

and

$$\widetilde{\mathbf{S}}_w = \frac{1}{2}\sum_{i,j=1}^{n} \widetilde{w}_{ij}^w \left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \tag{2.44}$$

respectively, where

$$\widetilde{w}_{ij}^b = \begin{cases} w_{ij}(1/n - 1/n_t) & \text{if } \mathbf{x}_i,\, \mathbf{x}_j \in C_t \\ w_{ij}/n & \text{otherwise} \end{cases} \tag{2.45}$$

$$\widetilde{w}_{ij}^w = \begin{cases} w_{ij}/n_t & \text{if } \mathbf{x}_i,\, \mathbf{x}_j \in C_t \\ 0 & \text{otherwise} \end{cases} \tag{2.46}$$

where w_{ij} is the ijth element of the pairwise weight matrix $\mathbf{W}$, which is a similarity/closeness measure based on features. LFDA is therefore a method of incorporating LPP into FDA.

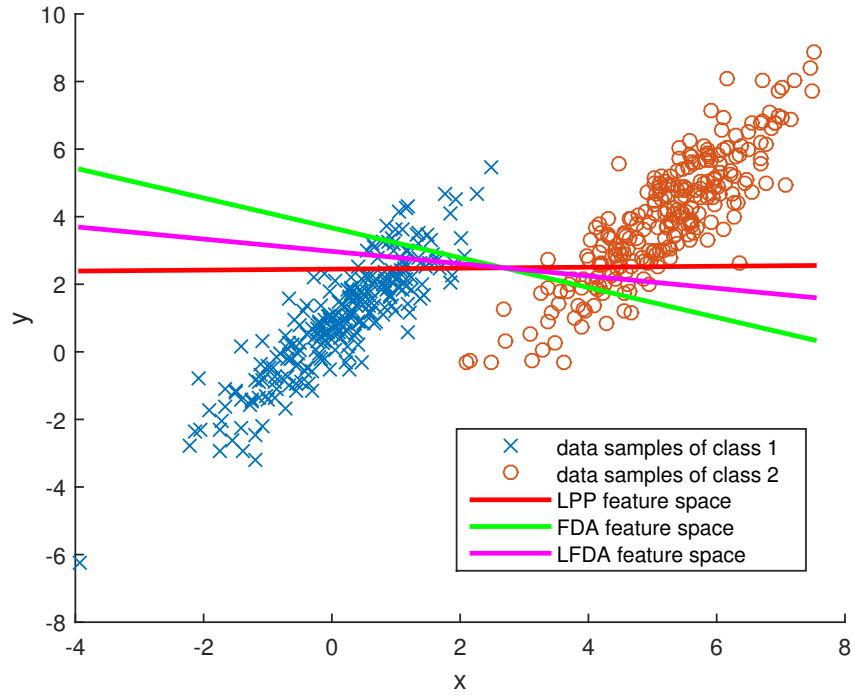
An example of embedding learning that compares the learning results of LFDA, FDA and LPP is shown in Figure 2.5a. The data samples of two different classes are generated from two multivariate normal distributions, with different means $(0.130, 0.870)$ and $(5.130, 3.870)$ while a common covariance $[1, 1.5; 1.5, 3]$. Compared to the projections of the data samples in the LPP and FDA feature space, those in the LFDA feature space (i.e. the magenta line shown in Figure 2.5a) achieve a good grouping and separation of the respective intraclass and interclass data samples, while preserving the original local neighbour information. To visually compare the local structures in the original, LPP, FDA and LFDA feature space, the proximity matrices are computed based on the Gaussian Weight listed in eq. (A.5). The resultant proximity matrices are displayed in Figure 2.6, where the data samples are ordered to ensure that those of the intraclass appear consecutively when plotted. From Figure 2.6, it can be seen that only FDA cannot preserve the local structure present in the original feature space. As a result, although LFDA has a less discriminate power, it can achieve a good balance between discriminating data samples and persevering local structures.

LFDA imposes the same optimisation problem as FDA to determine its projection function, as:

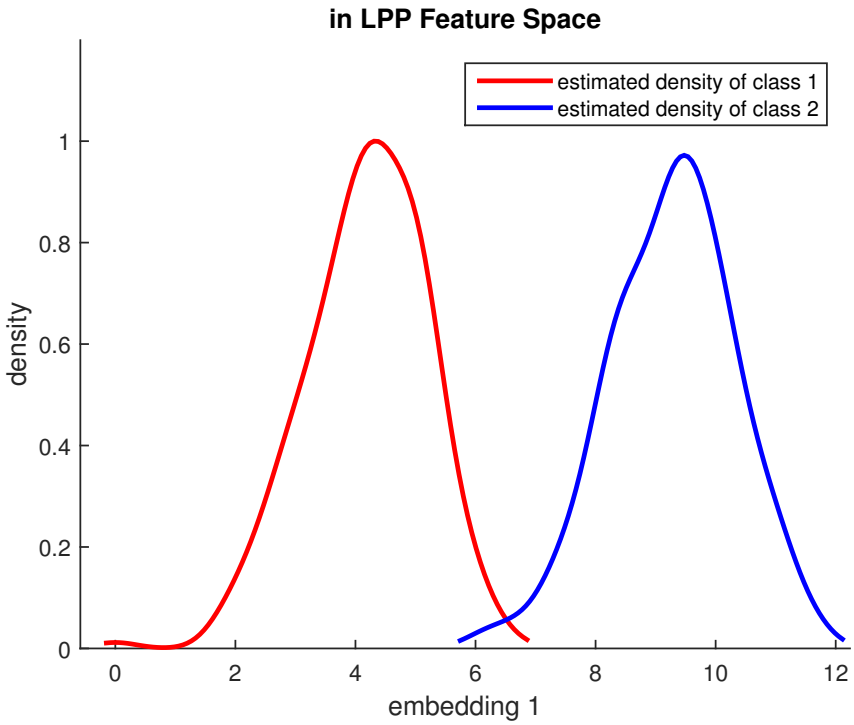
$$\underset{\mathbf{V}\in R^{d\times k}}{\arg\max} \frac{\operatorname{tr}\left[\mathbf{V}^T\widetilde{\mathbf{S}}_b\mathbf{V}\right]}{\operatorname{tr}\left[\mathbf{V}^T\widetilde{\mathbf{S}}_w\mathbf{V}\right]} \tag{2.47}$$

Similarly, the optimal projection matrix $\mathbf{V}^*$ can be obtained, with its columns $[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$ being the k generalised eigenvectors of $\widetilde{\mathbf{S}}_b$ and $\widetilde{\mathbf{S}}_w$, corresponding to the k largest generalised eigenvalues $[\lambda_1, \lambda_2, \ldots, \lambda_k]$. Through the determined mapping/projection function, LFDA makes nearby data samples from the same class stay close and all ones

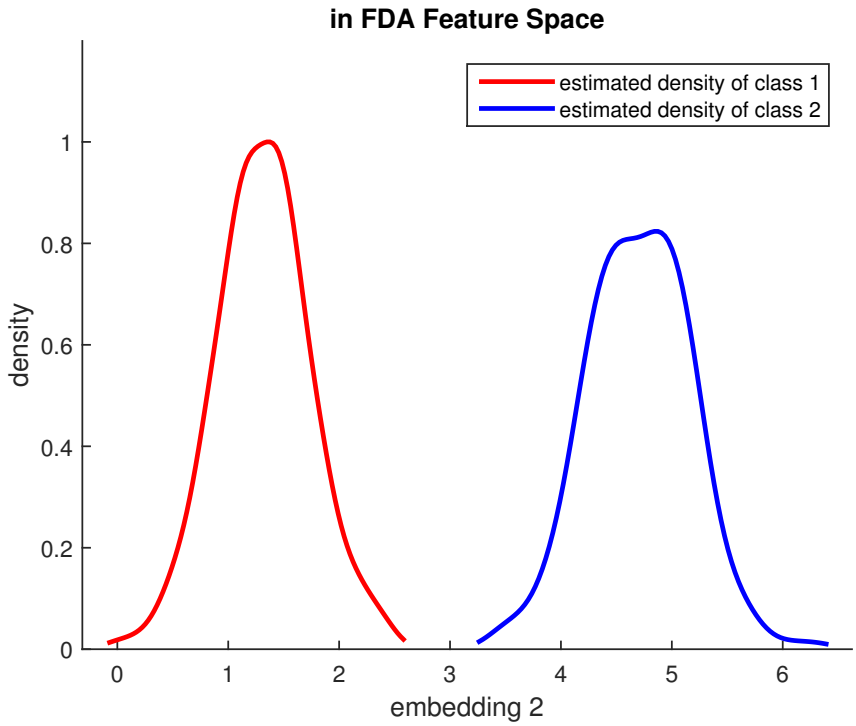
from different classes move apart. Because data samples from the same class that are far apart are not forced to be close, LFDA tends to preserve the local structures and has little effect on longer range connections [57].



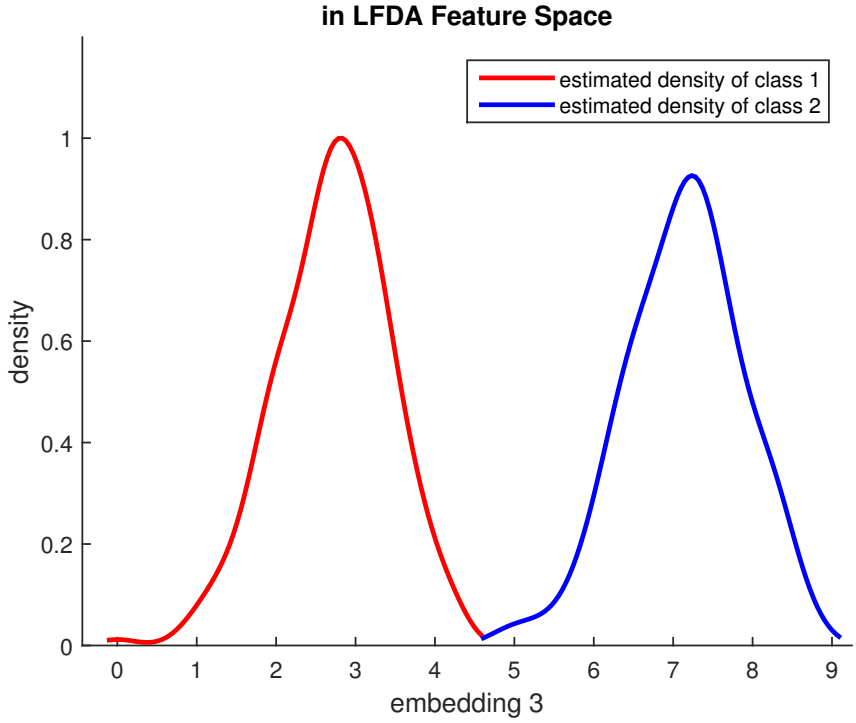
(a)



(b) the estimated densities of the projected data samples in the LPP feature space

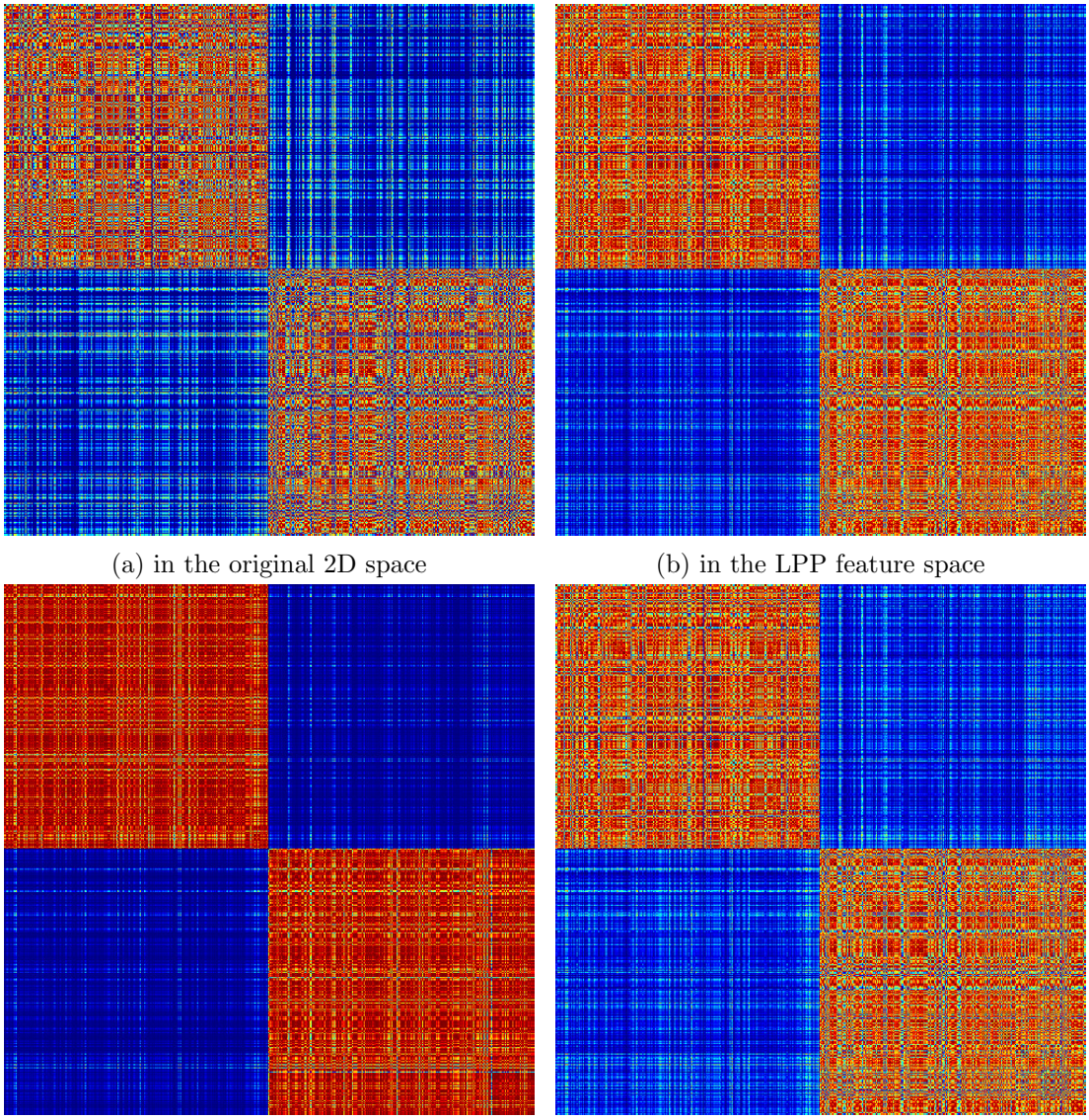


(c) the estimated densities of the projected data samples in the FDA feature space



(d) the estimated densities of the projected data samples in the LFDA feature space

Figure 2.5: An example of embedding learning - LFDA versus FDA versus LPP: (a) the projections of the original set in the LFDA feature space possess the discriminant power, while preserving the local structures.



(a) in the original 2D space

(b) in the LPP feature space

(c) in the FDA feature space

(d) in the LFDA feature space

Figure 2.6: Comparison of the local structures in the original, LPP, FDA and LFDA feature space: compared to the local structure in the original 2D feature space, that in the resultant LFDA feature space has almost the same structure, which demonstrates that LFDA is capable of preserving the local structures.

Marginal Fisher Analysis Marginal Fisher Analysis (MFA) [52] is another variant of FDA, which introduces local structures by formulating two graphs. It defines an intrinsic graph for intraclass (i.e. within-class) compactness, and a penalty graph for interclass (i.e. between-class) separability. In the intrinsic graph, a connection exists between two data samples, if one is among the k_1-NN of the other, and simultaneously both of them should belong to the same class. In the penalty graph, for each class, two data samples are connected, if one is in this class, while the other is among its k_2-NN and additionally from a different class. Compared to LFDA, MFA works in a different

way when dealing with interclass data samples: it keeps the neighbouring data samples from different classes far away.

Discriminative Locality Alignment Discriminative Locality Alignment (DLA) [53] targets to achieve the same goal as MFA, but it utilises a criterion similar to MMC, which results in the advantage of avoiding the small sample size problem. It uses the distances between each data sample and its intraclass k_1-NN for the measure of the within-class information, while those between each data sample and its interclass k_2-NN for the measure of the between-class information. Additionally, a scaling factor β in the range of 0 to 1 inclusive is employed to unify different measures of the within-class and between-class information [53].

Discriminant Neighbourhood Embedding

Discriminant Neighbourhood Embedding (DNE) [54] operates like OLPP, but incorporates the label information into a pairwise weight matrix $\mathbf{W}$ that is defined in OLPP. An example of embedding learning that compares the learning results of DNE, LFDA and FDA is shown in Figure 2.7a. The data samples of two different classes are generated from two multivariate normal distributions, with different means $(0.756, 0.228)$ and $(5.756, 3.228)$ while a common covariance $[1, 1.5; 1.5, 3]$. The resultant DNE feature space (i.e. the magenta line show in Figure 2.7a) is very similar to the resultant LFDA feature space (i.e. the green line shown in Figure 2.7a).
It replaces the weight matrix of OLPP with the following:

$$w_{ij} = \begin{cases} +1 & \text{if } \mathbf{x_j} \text{ is the intraclass } k\text{-NN of } \mathbf{x_i} \\ -1 & \text{if } \mathbf{x_j} \text{ is the interclass } k\text{-NN of } \mathbf{x_i} \\ 0 & \text{otherwise} \end{cases} \tag{2.48}$$

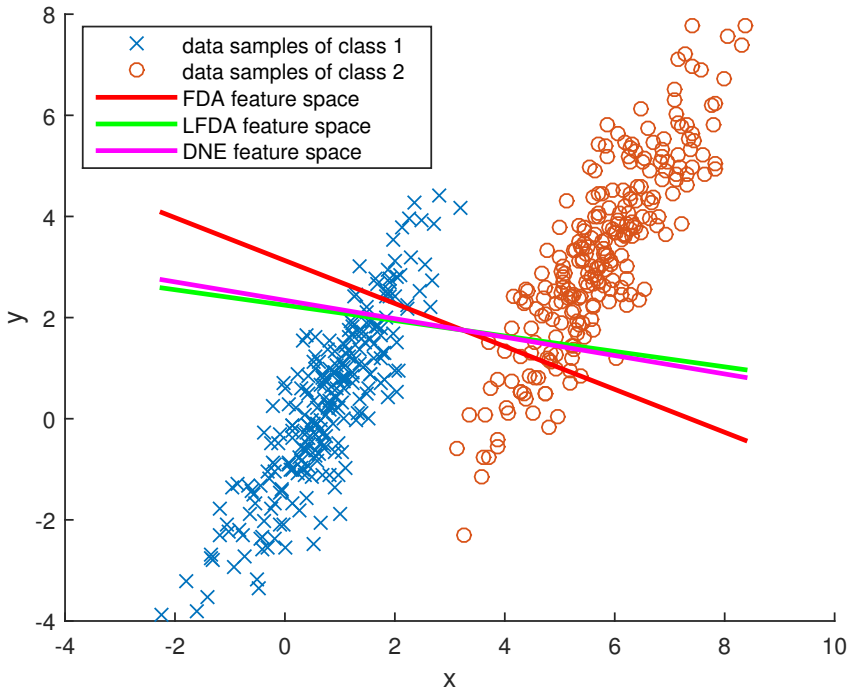
Then, it uses the same optimisation problem as that in OLPP (2.26), to determine the $d \times k$ orthogonal projection matrix $\mathbf{V}$, as:

$$\underset{\mathbf{V} \in R^{d \times k}, \mathbf{V}^T\mathbf{V} = \mathbf{I}_{k \times k}}{\arg\min} \operatorname{tr}\left[\mathbf{V}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{V}\right] \tag{2.49}$$

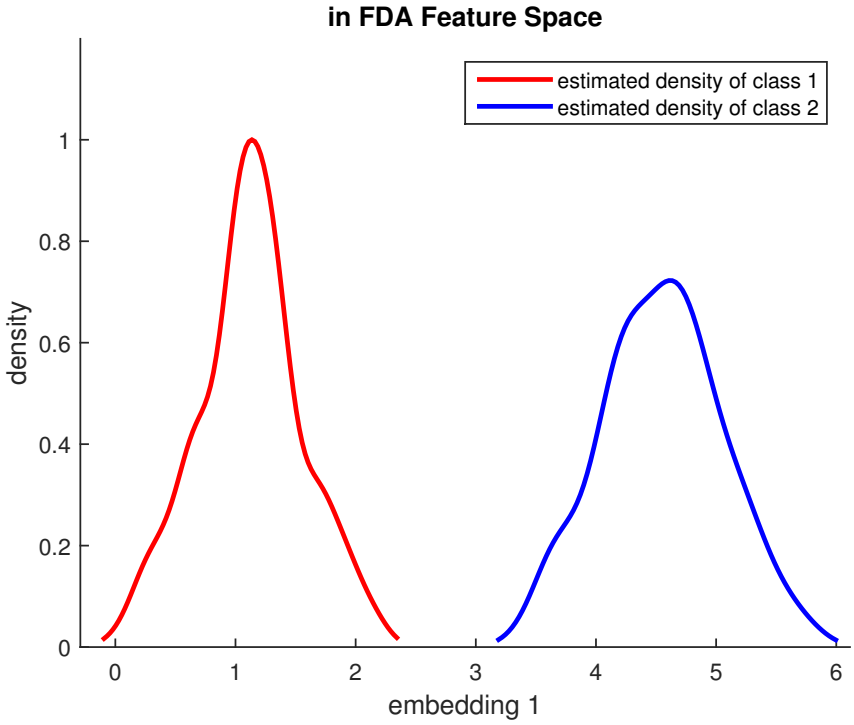
Using this, the optimal projection matrix $\mathbf{V}^*$ can be obtained, with its columns $[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$ are the k standard eigenvectors of $\mathbf{X}^T\mathbf{L}\mathbf{X}$, corresponding to the k smallest standard eigenvalues $[\lambda_1, \lambda_2, \ldots, \lambda_k]$. Through the determined mapping/projection function, DNE keeps neighbouring data samples from the same class squeezed together, but those from different classes separated as far as possible [54]. The projection results are similar to that of MFA and DLA [14].

Repulsion Orthogonal Locality Preserving Projections Repulsion Orthogonal Locality Preserving Projections (OLPP-R) [60] is another modified version of OLPP, which introduces a repulsion graph $\mathbf{L}_r$ and a new Laplacian graph $\mathbf{L}_l$ based on the label

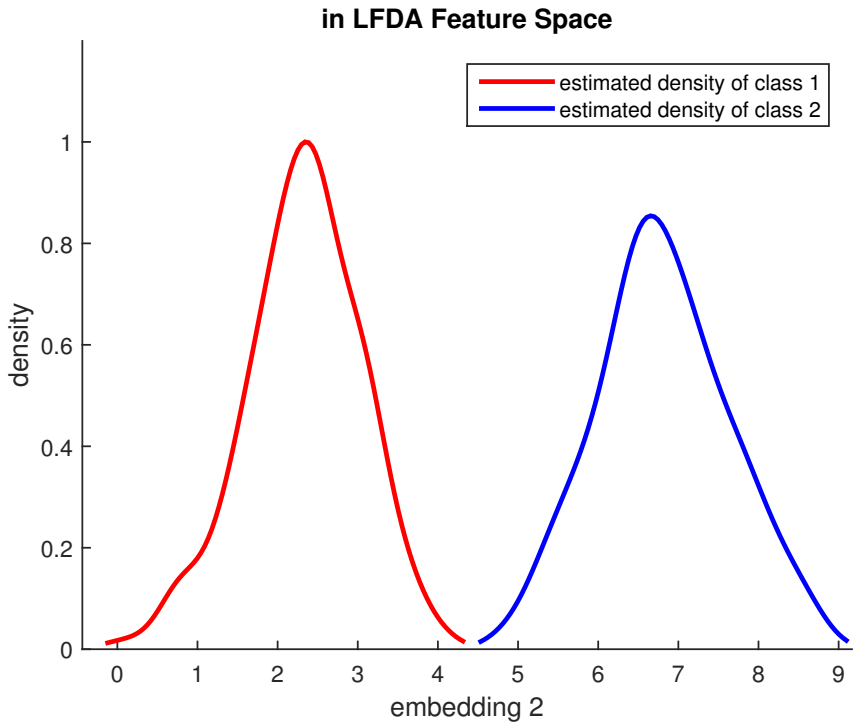
information. The repulsion graph $\mathbf{L}_r$ is formed from $\mathbf{L}_r = \mathbf{D}_r - \mathbf{W}_r$, where the pairwise weight matrix of repulsion $\mathbf{W}_r$ is derived from the data samples from different classes, where one is among the k-NN of the other, and vice versa. While for the labelled



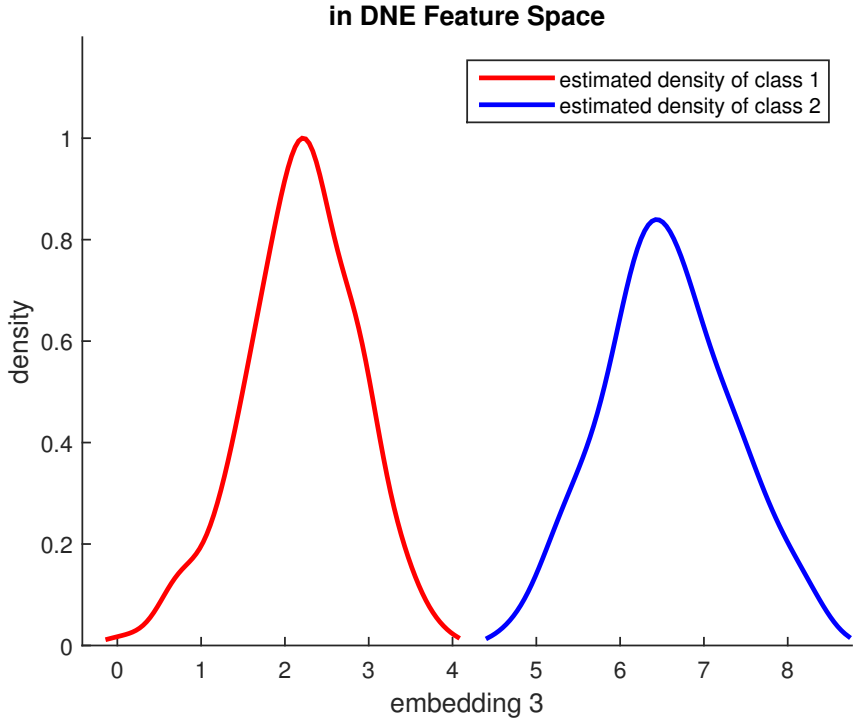
(a)



(b) the estimated densities of the projected data samples in the FDA feature space



(c) the estimated densities of the projected data samples in the LFDA feature space



(d) the estimated densities of the projected data samples in the DNE feature space

Figure 2.7: An example of embedding learning - DNE versus LFDA versus FDA: (a) the resultant feature space learned by DNE is almost the same as that learned by LFDA (i.e. DNE is capable of discriminating data samples while preserving local structures).

Laplacian graph $\mathbf{L}_l$, it is a graph formed from $\mathbf{L}_l = \mathbf{D}_l - \mathbf{W}_l$, where the pairwise weight matrix of Laplacian $\mathbf{W}_l$ is constructed by the data samples in the same class. Then a linear combination of $\mathbf{L}_r$ and $\mathbf{L}_l$ is made to form the objective function, as:

$$\underset{\mathbf{V}\in R^{d\times k}, \mathbf{V}^T\mathbf{V}=\mathbf{I}_{k\times k}}{\arg\min} \operatorname{tr}\left[\mathbf{V}^T\mathbf{X}^T(\mathbf{L}_l - \beta\mathbf{L}_s)\mathbf{X}\mathbf{V}\right] \tag{2.50}$$

where $\beta > 0$ is a penalty parameter.

Repulsion Orthogonal Neighbourhood Preserving Projections Repulsion Orthogonal Neighbourhood Preserving Projections (ONPP-R), which are also proposed in [60], works almost the same as OLPP-R, but it is based on introducing a repulsion graph and the label information to modify the LLE style weight used in ONPP (appendix A.1). ONPP-R aims to keep the local structures within each class, and make the neighbouring data samples of interclass to move apart.

Supervised Orthogonal Neighbourhood Preserving Projections Supervised Orthogonal Neighbourhood Preserving Projections (SONPP) [45] is an implementation of ONPP in a supervised setting. A connection between two data samples $\mathbf{x}_i$ and $\mathbf{x}_j$ exists only when $\mathbf{x}_i$ and $\mathbf{x}_j$ are from the same class. In other words, the LLE style weight is computed based on all the data sample in the same class. It results of projections that carry both geometric (structure within each class) and discriminating (class label) information.

Discriminative Orthogonal Neighbourhood Preserving Projections Discriminative Orthogonal Neighbourhood Preserving Projections (DONPP) [63] is a supervised version of ONPP, which is built based on SONPP [45]. Like SONPP, it assumes that each data sample can be reconstructed from the remaining ones in the same class, and additionally, it expects that neighbouring data samples from different classes are as far away as possible.

2.4 Semi-Supervised Learning Techniques

Usually, the process of labelling data samples needs to be done by hand (i.e. by a person), which is time consuming and can be error prone. Sometimes, because the cost of associating each data sample with a label is high (e.g. the labelling process requires experts and/or special devices), there is only a portion of (typically, a small amount of) the data samples that are labelled. To address this special type of dataset, a group of algorithms called semi-supervised learning have been proposed, where both labelled and unlabelled data samples are used to determine mapping functions. Some commonly used ones in the direct approach group are listed, each of which combines an unsupervised learning technique with a supervised one.

Semi-supervised Local Fisher discriminant analysis

SEmi-supervised Local Fisher discriminant analysis (SELF) [69] is a method that combines PCA and LFDA, where it utilises PCA for both labelled and unlabelled data samples, and LFDA for the labelled ones alone. An example of embedding learning that compares the learning results of SELF, LFDA and PCA is shown in Figure 2.8a. Two sets of data samples were generated from two different multivariate normal distributions, with different means $(0.551, 0.848)$ and $(5.551, 3.848)$ while a common covariance $[1, 1.5; 1.5, 3]$. Each set only has a small number of labelled data samples, with 5% of the data samples labelled. The SELF feature space (i.e. the magenta line shown in Figure 2.8a) has the most discriminant power, compared to the PCA and FDA feature spaces.

SELF uses the same optimisation problem as in FDA and LFDA, but with the between-class and within-class scatter matrices $\mathbf{S}_b$ and $\mathbf{S}_w$ being redefined to be:

$$\mathbf{S}_b = (1-\beta)\mathbf{S}_b^l + \beta\mathbf{S}_t \tag{2.51}$$

$$\mathbf{S}_w = (1-\beta)\mathbf{S}_w^l + \beta\mathbf{I}_{d\times d} \tag{2.52}$$

where $\mathbf{S}_b^l$ and $\mathbf{S}_w^l$ are the respective between-class and within-class scatter matrices calculated from the labelled data samples, according to LFDA. $0 < \beta < 1$ is a trade-off parameter that inherits the properties and characteristics of both LFDA and PCA, and $\mathbf{S}_t$ is the total scatter matrix of all the data samples that is defined by PCA, as:

$$\mathbf{S}_t = \frac{1}{2}\sum_{i,j=1}^{n}\frac{1}{n}\left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \tag{2.53}$$

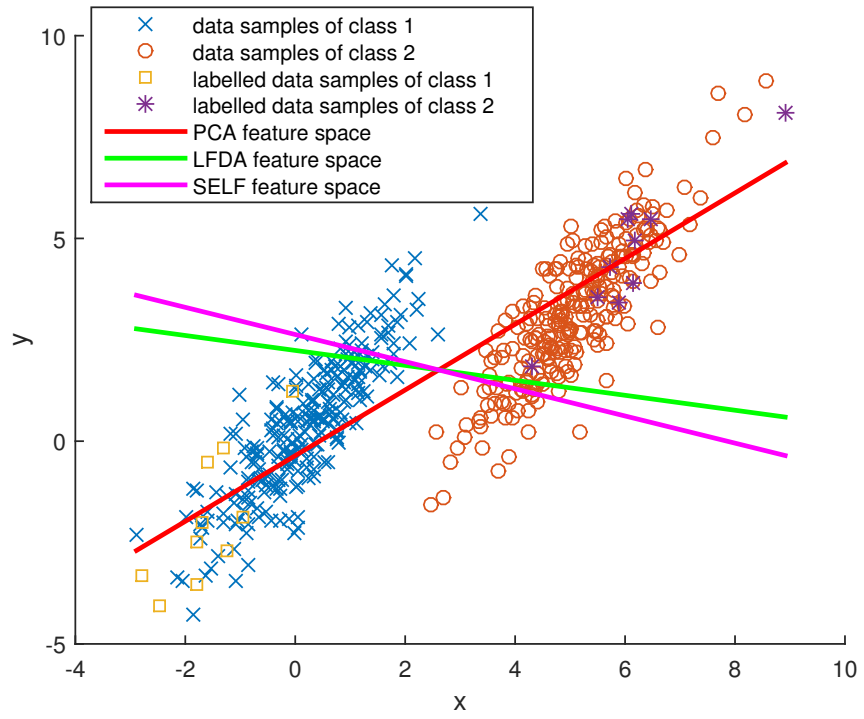
The optimal projection matrix $\mathbf{V}^*$ is then obtained in the same way as FDA and LFDA, with its columns $[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$ being the k generalised eigenvectors of $\mathbf{S}_b$ and $\mathbf{S}_w$, corresponding to the k largest generalised eigenvalues $[\lambda_1, \lambda_2, \ldots, \lambda_k]$.

Semi-Supervised Fisher Discriminant Analysis Semi-Supervised Fisher Discriminant Analysis (SSFDA) [68] is another method of semi-supervised learning, which combines OLPP with FDA. It employs the same maximisation problem that is in FDA, but with the within-class scatter matrix $\mathbf{S}_w$ redefined to be:

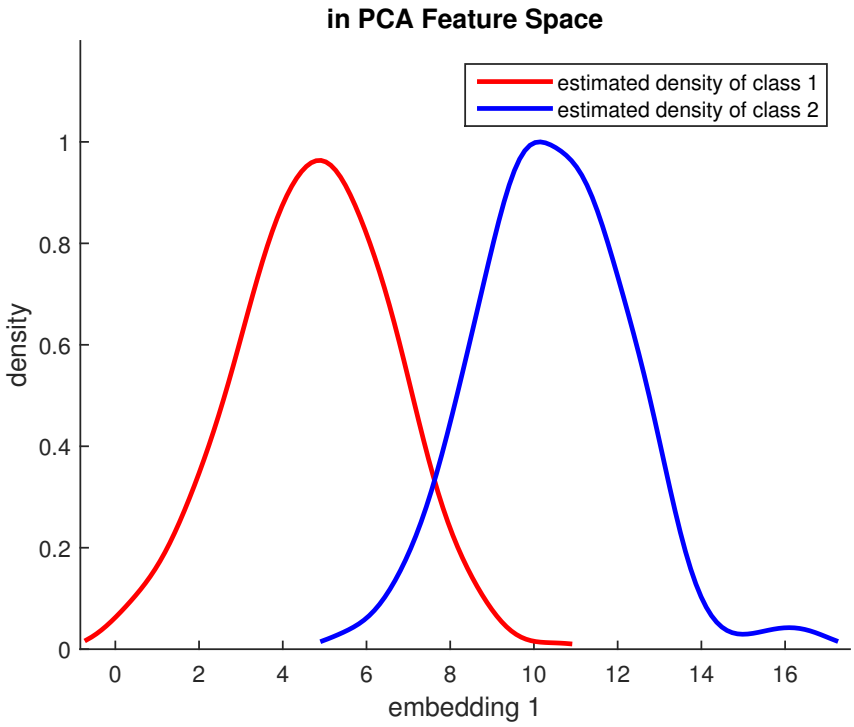
$$\mathbf{S}_w = \mathbf{S}_w^l + \beta_1\mathbf{X}^T\mathbf{L}\mathbf{X} + \beta_2\mathbf{I}_{d\times d} \tag{2.54}$$

where $\mathbf{S}_w^l$ is the within-class scatter matrix computed from the labelled data samples, based on FDA, while $\mathbf{X}^T\mathbf{L}\mathbf{X}$ are computed from all the data samples, according to OLPP. $\beta_1 > 0$ and $\beta_2 > 0$ are parameters that control the balance of the terms. Here, $\mathbf{X}^T\mathbf{L}\mathbf{X}$ is added as a regularisation term [68], which is introduced to prevent overfitting. It can be computed based on the Laplacian matrix of a neighbouring graph, where it makes the use of the information provided by both labelled and unlabelled data samples.

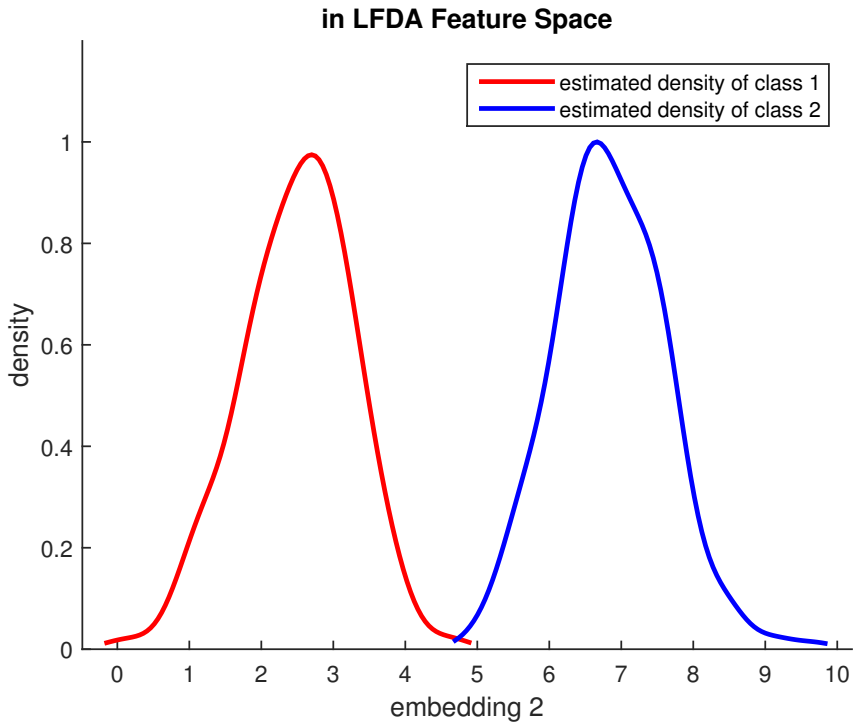
Semi-Supervised Maximum Margin Criterion Semi-Supervised Maximum Margin Criterion (SSMMC) is another semi-supervised method proposed in [68]. It is based on the use of MMC to avoid the small sample size problem. It redefines the within-class



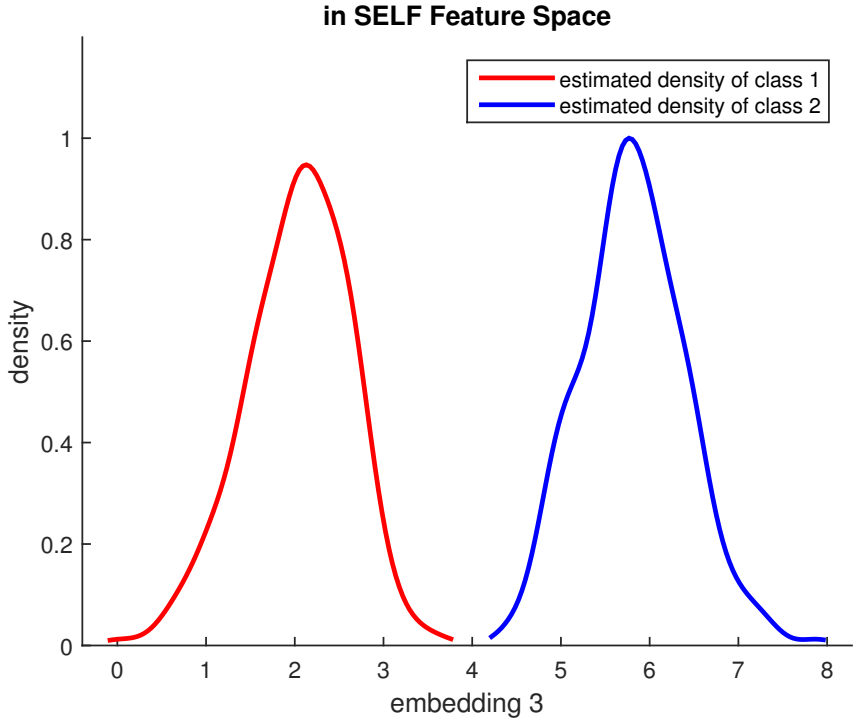
(a)



(b) the estimated densities of the projected data samples in the FDA feature space



(c) the estimated densities of the projected data samples in the LFDA feature space



(d) the estimated densities of the projected data samples in the DNE feature space

Figure 2.8: An example of embedding learning - SELF versus LFDA versus PCA: (a) the projections of the original 2D data in the SELF feature space possess the most discriminant ability.

scatter matrix $\mathbf{S}_w$ in the maximisation problem of MMC to be:

$$\mathbf{S}_w = \beta_1 \mathbf{S}_w^l + \beta_2 \mathbf{X}^T \mathbf{L} \mathbf{X} \tag{2.55}$$

where $\mathbf{S}_w^l$ is the within-class scatter computed from the labelled data samples, based on MMC, and $\beta_1 > 0$ and $\beta_2 > 0$ are parameters that control the balance of the terms.

2.5 Learning Techniques for Multi-Label Classification

Multi-label classification refers to the classification problem where each data sample may be assigned two or more target labels. The supervised and semi-supervised techniques described above provide single label classification cannot be used in these situations. Various algorithms have been proposed for multi-label classification. Some commonly used ones are listed below.

Partial Least Squares

A direct way to approach embedding learning for multi-label classification is to get an optimal statistical criterion between the resultant embeddings and the corresponding label information. Partial Least Squares (PLS) [76,77] aims to maximise the covariance between embedding results and the corresponding label information. It is designed to look for directions that are best at distinguishing data samples possessing different labels. It gets the optimal projection function by solving the following optimisation problem:

$$\underset{\mathbf{V} \in R^{d \times k}, \mathbf{V}^T \mathbf{V} = \mathbf{I}_{k \times k}}{\arg\max} \operatorname{tr}[\mathbf{V}^T \mathbf{X}_c^T \mathbf{Y}_c \mathbf{Y}_c^T \mathbf{X}_c \mathbf{V}] \tag{2.56}$$

where $\mathbf{X}_c$ and $\mathbf{Y}_c$ are the centred training set and label information matrices, respectively, they are computed using:

$$\mathbf{X}_c = \left(\mathbf{I}_{n \times n} - \frac{1}{n} \mathbf{1}_{n \times n} \right) \mathbf{X} \tag{2.57}$$

$$\mathbf{Y}_c = \left(\mathbf{I}_{n \times n} - \frac{1}{n} \mathbf{1}_{n \times n} \right) \mathbf{Y} \tag{2.58}$$

Orthonormalised Partial Least Squares Orthonormalised Partial Least Squares (OPLS) [76, 78] is a variant of PLS that modifies the original constraint of PLS to be $\mathbf{V}^T \mathbf{X}_c^T \mathbf{X}_c \mathbf{V}$. It results in each dimension of the generated embeddings being orthonormal to each other.

Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) [79, 80] is another method that has been proposed on the basis of utilising the optimisation of a statistical measure. It operates like

PLS, but instead, it attempts to maximise the correlation coefficient between the embedding results and the corresponding label information. It is a method that extracts features based on making the use of all the label information of the same data sample. It gets the optimal projection function by solving the following optimisation problem:

$$\underset{\mathbf{V}\in R^{d\times k},\mathbf{V}^T\mathbf{X}_c^T\mathbf{X}_c\mathbf{V}=\mathbf{I}_{k\times k}}{\arg\max} \operatorname{tr}[\mathbf{V}^T\mathbf{X}_c^T\mathbf{Y}_c(\mathbf{Y}_c^T\mathbf{Y}_c)^{-1}\mathbf{Y}_c^T\mathbf{X}_c\mathbf{V}] \tag{2.59}$$

Regularised Canonical Correlation Analysis A regularised version of CCA, regularised Canonical Correlation Analysis (rCCA) [80,81] has also been proposed. It adds a regularisation parameter $\lambda > 0$ into the original constraint of CCA, and the modified constraint becomes $\mathbf{V}^T(\mathbf{X}_c^T\mathbf{X}_c + \lambda\mathbf{I}_{d\times d})\mathbf{V} = \mathbf{I}_{k\times k}$. After this constraint modification, the singularity of $\mathbf{X}_c^T\mathbf{X}_c$ (in the original constraint) due to the problem of small sample size can be avoided, which will prevent overfitting.

2.6 Techniques for Nonlinear Extension

Although these nonlinear methods are not considered further in this thesis, they are included for completeness, and linear techniques can be extended using the methods discussed. Compared to the nonlinear embedding learning model, the above linear model is simpler, more efficient and straightforward when dealing with out-of-sample extensions, i.e. generating embeddings for new test data samples. However, for some set of data samples, the actually meaningful structure lies on a nonlinear manifold. In such case, the techniques for linear embedding learning can on longer correctly model the variability within this kind of set. To address this problem, some techniques have been developed to extend the techniques for linear embedding learning to nonlinear learning, thereby increasing their powers.

Kernel Trick

The standard approach to achieve a nonlinear extension of a linear method is to apply the kernel trick [96–98], which has been extensively used in the context of Support Vector Machines (SVMs) [99,100]. Through the use of a kernel that defines inner products between pairs of data samples, a set of data samples in its original features space is transformed to a high dimensional space. This results in avoiding the explicit relationship between the original set and the resultant embeddings learned by the techniques on linear embedding learning. The high dimensional feature space, which is known as the kernel-induced feature space, is usually implicit, as it is only based on a simple measure of the inner product.

Let the function $\phi_k : R^d \to H$ donate a mapping function that transforms the training set $\mathbf{X}$ in its original feature space to the kernel-induced feature space H, then

the resultant set of data samples in the kernel-induced feature space H is denoted by the matrix $\mathbf{\Phi}_k$, as:

$$\mathbf{\Phi}_k = [\phi_k(\mathbf{x}_1), \phi_k(\mathbf{x}_2), \ldots, \phi_k(\mathbf{x}_n)]^T \tag{2.60}$$

Let an $n \times n$ matrix $\mathbf{K}$ denote the results of the inner products between the pairs of the data samples in the kernel-induced feature space H, with its ijth element k_{ij} being obtained through:

$$k_{ij} = \phi_k(\mathbf{x}_i)^T \phi_k(\mathbf{x}_j) \tag{2.61}$$

By Mercer's theorem (i.e. any positive semidefinite matrix is the Hermitian matrix of inner products), $\mathbf{K}$ must be Positive Semi-Definite (PSD) [101].

The objective is then to determine a transformation function $\widetilde{\mathbf{V}} = [\widetilde{\mathbf{v}}_1, \widetilde{\mathbf{v}}_2, \ldots, \widetilde{\mathbf{v}}_k]$, so that the training set $\mathbf{\Phi}_k$ can be mapped to a new feature space of dimension k. The classical kernel trick links each projection direction $\widetilde{\mathbf{v}}_j$ $(j = 1, \ldots, k)$ to a linear combination of all the data samples in $\mathbf{\Phi}_k$ [15], as:

$$\widetilde{\mathbf{v}}_j = \sum_{i=1}^{n} \gamma_{ij} \phi_k(\mathbf{x}_i) \tag{2.62}$$

where γ_{ij} is the ijth element of the $n \times k$ coefficient matrix $\mathbf{\Gamma}$, so that $\widetilde{\mathbf{V}} = \mathbf{\Phi}_k^T \mathbf{\Gamma}$. Then, the embedding results of the training set $\mathbf{Z}$ can be obtained using:

$$\begin{aligned} \mathbf{Z} &= \mathbf{\Phi}_k \widetilde{\mathbf{V}} \\ &= \mathbf{\Phi}_k \mathbf{\Phi}_k^T \mathbf{\Gamma} \\ &= \mathbf{K}\mathbf{\Gamma} \end{aligned} \tag{2.63}$$

Finally, the resultant embeddings of the corresponding test set $\widetilde{\mathbf{Z}}$ can be computed by:

$$\widetilde{\mathbf{Z}} = \widetilde{\mathbf{K}}\mathbf{\Gamma} \tag{2.64}$$

where $\widetilde{\mathbf{K}}$ is a $m \times n$ matrix that defines the inner product results between the test and training sets in the kernel-induced feature space H, with its ijth element $\widetilde{k}_{ij}$ being obtained through:

$$\widetilde{k}_{ij} = \Phi_k(\widetilde{\mathbf{x}}_i)^T \Phi_k(\mathbf{x}_j) \tag{2.65}$$

where $\Phi_k(\widetilde{\mathbf{x}}_i)$ is the ith row of the matrix $\widetilde{\mathbf{\Phi}}_k$, as:

$$\widetilde{\mathbf{\Phi}}_k = [\phi_k(\widetilde{\mathbf{x}}_1), \phi_k(\widetilde{\mathbf{x}}_2), \ldots, \phi_k(\widetilde{\mathbf{x}}_m)]^T \tag{2.66}$$

which denotes the mapping results of the test set $\widetilde{\mathbf{X}}$ in the kernel-induced feature space H.

Relation Features

Relation features are generated based on a similarity measure $\Phi_r(\cdot,\cdot)$ between all the pairs of data samples [14,15]. Employing relation features to adapt a linear embedding learning technique for nonlinear learning is motivated by the kernel trick. Comparing (2.5) and (2.63), it can be found that they share a similar form. But in (2.63), the inner product results $\mathbf{K}$ is used instead of the training set $\mathbf{X}$, and the coefficient $\mathbf{\Gamma}$ is used instead of the projection $\mathbf{V}$. Since the inner product results $\mathbf{K}$ are actually a way to measure the similarity between all the pairs of the data samples in the the training set $\mathbf{X}$, then the relation features (i.e. the result of a similarity measure) can be used instead of the inner product results $\mathbf{K}$. Based on the fact that the inner product results $\mathbf{K}$ which do not satisfy Mercer's condition may still perform reasonably, if they at least approximate the intuitive idea of similarity measure between pairs of data samples, an arbitrary similarity measure can be employed to generate relation features. Then, for the training set $\mathbf{X}$, the relation features can be denoted by an $n \times n$ matrix $\mathbf{F}$, with its ij element f_{ij} being computed as:

$$f_{ij} = \Phi_r(\mathbf{x}_i, \mathbf{x}_j) \tag{2.67}$$

Applying relation features $\mathbf{F}$ to generate embeddings can provide two advantages. One is reducing the computational complexity, as the relation features $\mathbf{F}$ possesses a dimension of n while the training set $\mathbf{X}$ has a dimension of d, and usually $n \ll d$ [102]. The other is in capturing interactions between the data samples, which can be used to help discover the nonlinear structures included in the training set $\mathbf{X}$ [14]. Since each data sample can be adequately represented by a similarity or dissimilarity measure with other ones [103]. The effectiveness of using the relation features instead of the original features as the input to machine learning algorithms has been demonstrated in many previous works [104, 105]. The work in [106] also indicates that embeddings computed based on the relation features can achieve similar discriminant performance to those generated based on the original features.

Chapter 3

Image Registration & Its Application in 3D Reconstruction

3.1 Background

Image registration refers to the process of discovering the corresponding relationships between pixel locations in two images so that they can be geometrically aligned, based on certain degrees of overlap between them. Additionally, a function for interpolating intensity value is employed to assign appropriate intensity values to pixel locations in the transformed image, so that the transformed images can be properly aligned to the reference one (i.e. the spatial error is minimised). A very wide range of image registration techniques have been developed and they are employed in a wide variety of applications, including medical imaging and remote sensing [17, 18, 20], video compression [107–109], video summarisation [110–113], photographic mosaic [114], panoramic photography [115–120], as well as image stitching [121–127], and so on. There has been a rapid development in the image acquisition technology in the past years, so that it becomes necessary to automatically and reliably register images with a higher quality (e.g. frame size, frame rate, colour and dynamic range) [18].

Usually, the inputs to the registration process are two images of the same scene. One is called the reference image, which is kept unchanged and used as the basis for the transformation; the other is called the sensed image, which is going to be geometrically transformed, so that it is spatially aligned with the reference one.

Let two 2D arrays of a given size I_R and I_S denote arbitrary two 2D images of the same scene, with I_R and I_S being the reference and sensed images, respectively. Then their respective intensity values at the pixel location (x, y) are expressed as $I_R(x, y)$ and $I_T(x, y)$, which are related by:

$$I_R(x, y) = g(I_S(f_x(x, y), f_y(x, y))) \tag{3.1}$$

where $f_x(\cdot, \cdot)$ and $f_y(\cdot, \cdot)$ are 2D spatial relationship functions, which map the pixel location (x, y) in the original spatial coordinate to the pixel location (x', y') in a new

spatial coordinate, as:

$$(x', y') = (f_x(x, y), f_y(x, y)) \tag{3.2}$$

and $g(\cdot)$ is an intensity interpolation function, which assigns an appropriate intensity value to the pixel location (x', y'). Typically, both I_R and I_S are assumed to be acquired under the assumption of an ideal pinhole camera, which is a first order approximation of the mapping result from a 3D scene to a 2D image [128]. In general, images acquired under this assumption are good descriptions of the original scene, where it ignores geometric distortions or blurring of unfocused objects caused by lenses and/or finite sized apertures [128].

Based on the input images obtained by different acquisition manners and the final output, the registration process can be divided into four main analysis groups [18]. The first group is multiview (i.e. different viewpoints) [18], which obtains a new representation with higher resolution based on images of the same scene acquired from different viewpoints. For example, area mosaicing in remote sensing, wide-angle view in panoramic photography, and 3D shape reconstruction in computer vision [18]. The second group is multitemporal (i.e. different times) [18], where images of the same scene acquired at different times and often under different conditions are computed, to recognise and evaluate the changes. Example applications include landscape monitoring in remote sensing, motion tracking in computer vision, and health monitoring in medical imaging [18]. The third group is multimodal (i.e. different sensors) [18], which integrates information obtained form different source streams to obtain a scene representation with more details, based on images of the same scene acquired by different sensors. For instance, image fusion in remote sensing, and image combination in medical imaging [18]. The last group is scene to model registration [18], where images of a scene and a virtual model are registered, so that the acquired image is localised for comparison. Example application includes registering aerial satellite data into other GIS maps in remote sensing [18].

There is no universal method that is applicable to all registration tasks, due to the large variety of image acquisition processes and the various degradation types. Every method should take into account the geometric distortions, and the data characteristics, such as radiometric deformation and noise corruption. Nevertheless, the main steps needed to register images are as follows: detecting features, matching features, estimating transformation model, as well as transforming and resampling images. The existing techniques for image registration can be divided into two categories: area based and feature based, based on whether to directly use pixel intensity value as the features for matching. Area based registration techniques combine the direct comparison of the pixel intensity values of two images with optimisation techniques to estimate the parameters needed for the alignment of the two images. Feature based techniques find the distinctive features in two images and sequentially match the detected features to

establish the correspondence between two images.

Although the registration process includes two transformation steps (i.e. first a geometric transformation that modifies pixel locations, and then a photometric transformation that assigns intensity values), in this section the images are assumed to differ only by a geometric transformation, so that the corresponding mathematical expressions can be simplified. A geometric transformation can be described mathematically by an appropriate transformation model, which relates the spatial coordinate between two images. The techniques for photometric transformation will be examined separately. Image registration has a board range of applications in remote sensing, and computer vision [20]. For the application of 3D reconstruction techniques to 2D images, it is the fundamental part of the whole process.

3.2 Transformation Model

In general, there exist a number of possible transformation models that can describe different images. Both images, I_R and I_S, are assumed to have been obtained using an ideal pinhole camera. For simplicity, it will be assumed that the possible mathematical relationships that relate them are limited to the basic set of five 2D planar transformations, i.e. translation, Euclidean, similarity, affine and projective.

3.2.1 Pinhole Camera Model

The pinhole camera model [129,130] is the geometric and mathematical representation of the ideal pinhole camera, which is the simplest way to describe the acquisition of images. It assumes that the camera aperture is infinitely small (i.e. a point) and no lens is used to focus light. The geometry and mathematics of a pinhole camera are shown in Figure 3.1a. Let $P(-x_1, x_2, x_3)$ and $Q(y_1, -y_2)$ denote a point in the 3D world scene and the corresponding projection in the 2D image plane, respectively. In Figure 3.1a, it can be seen that there are two groups of similar triangles. One is obtained as seen from the X_2 axis, as shown in Figure 3.1b. $Q(y_1, -y_2)$ and $P(-x_1, x_2, x_3)$ are related through the following equation:

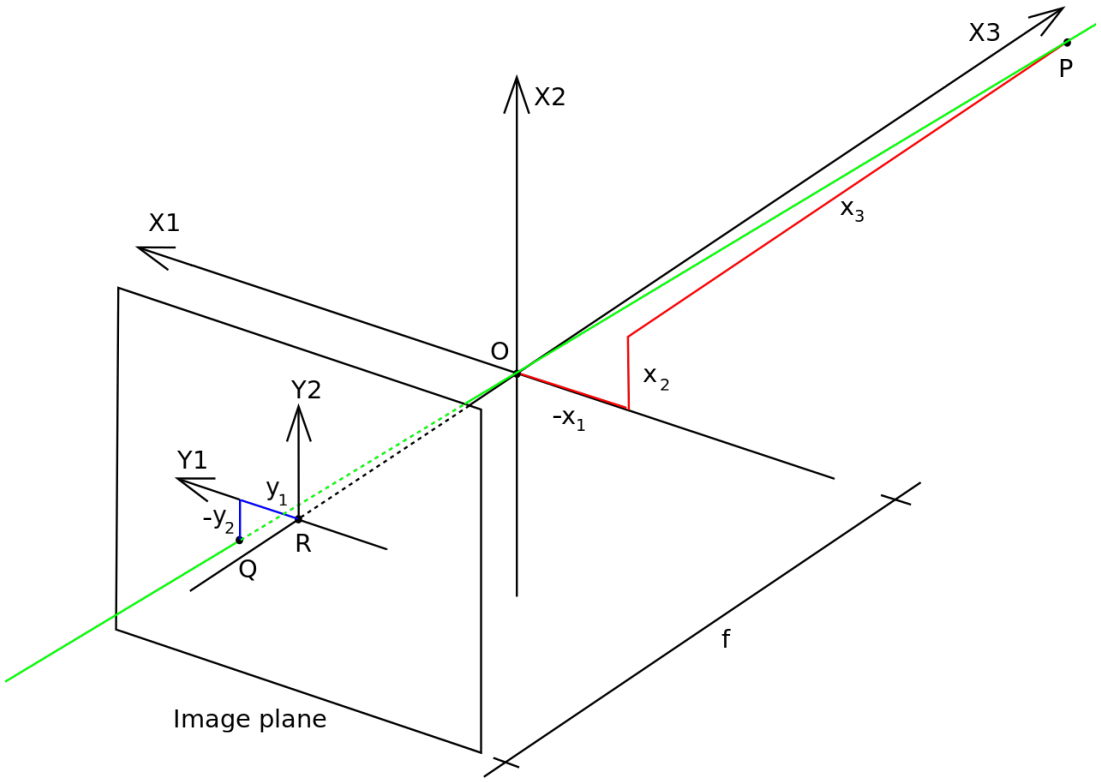
$$\frac{y_1}{f} = \frac{-x_1}{x_3} \tag{3.3}$$

The other is obtained as seen from the X_1 axis. Similarly, a relation equation can be obtained, as:

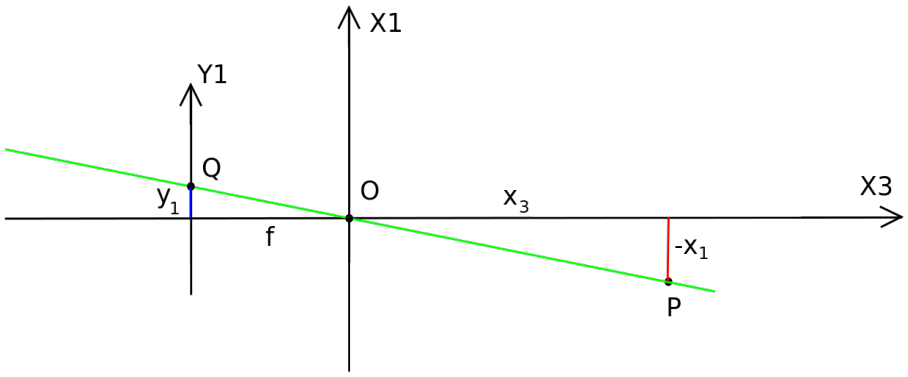
$$\frac{-y_2}{f} = \frac{x_2}{x_3} \tag{3.4}$$

where f is the focal length of the pinhole camera. eq. (3.3) and eq. (3.4) above can be merged as one matrix notation, as:

$$\lambda \mathbf{Y}_{\mathrm{2D}} = \mathbf{P}\mathbf{X}_{\mathrm{3D}} \tag{3.5}$$



(a) the geometric and mathematical representation of an ideal pinhole camera



(b) one group of similar triangles as seen from the X_2 axis

Figure 3.1: An example of an ideal pinhole camera.

where $\mathbf{Y}_{2D} = [y_1, y_2, 1]^T$ and $\mathbf{X}_{3D} = [x_1, x_2, x_3, 1]^T$ are the expressions in the homogeneous coordinates respectively, and $\lambda = -x_3/f$ is a scale factor, while $\mathbf{P} = [\mathbf{I}_{3\times3}, \mathbf{0}_{3\times1}]$ is a 3×4 matrix representing the projection relation between $\mathbf{Y}_{2D}$ and $\mathbf{X}_{3D}$ [128, 131].

Camera Calibration

The camera model above is an ideal case, which does not take into account either intrinsic or extrinsic parameters that could result from camera calibration [128,132,133]. In general, both intrinsic and extrinsic parameters parameter may exist, and then both internal and external calibrations should be incorporated into the projection relation

matrix **P**.

Intrinsic Parameters The lens is not explicitly modelled in the pinhole camera so the effect of the radial lens distortion [134] is not be taken into account. There are three main different types of parameters that would result from an internal calibration. The first type of parameters is α and β, which are the scaling factors in the respective Y_1 and Y_2 directions of the image plane (i.e. the aspect ratio of the image plane may not be unity). The second type is (u_0, v_0), which is the offset coordinate of the principal point (i.e. the image centre) that is the point where the optic axis intersects the image plane, since the principal point may not be at the origin of the image plane (usually the top left corner). The last one is the skew descriptor γ, which represents the skew information of the Y_1 and Y_2 axes of the image plane due to the fact that the image plane may not be exactly rectangular. The above listed parameters can be summarised into a 3×3 matrix, i.e. the camera intrinsic parameter $\mathbf{K}_{\text{cam}}$ matrix [128, 132, 133], as:

$$\mathbf{K}_{\text{cam}} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

Lens Distortion Although the pinhole camera model is a good approximation to the behaviour of most real cameras, it can be significantly improved upon when taking into account lens distortion. Lens distortion refers to the situation where image pixel locations are displaced from the position predicted by the ideal pinhole camera model. Its most common form is radial distortion, where the displacement away or toward the image centre is proportional to the corresponding radial distance [135]. Under the assumption that the principal point is the same as the centre of distortion, the correction of radial distortion can be achieved by [136, 137]:

$$\hat{y}_1 = y_1 + L_{\text{D}}(r) y_1 \tag{3.7}$$

$$\hat{y}_2 = y_2 + L_{\text{D}}(r) y_2 \tag{3.8}$$

where $(\hat{y}_1, \hat{y}_2)$ is the corrected position for the pixel location (y_1, y_2). $L_{\text{D}}(r)$ is the distortion function and can be approximated by using low-order polynomials as:

$$L_{\text{D}}(r) \approx l_{d1} r^2 + l_{d2} r^2 \tag{3.9}$$

where l_{d1} and l_{d2}, called the radial distortion parameters [114, 138], are considered to be camera intrinsic parameters, and $r = \sqrt{y_1^2 + y_2^2}$ is the radial distance.

Extrinsic Parameters The external calibrations are simpler and more straightforward. It aims to relate the world coordinate system, that includes the 3D scene, to the camera coordinate system that includes the 2D image plane. It consists of three

rotation angles θ_1, θ_2 and θ_3 around each of the three respective coordinate axes X_1, X_2 and X_3, where each angle results of a 3×3 rotation matrix, as $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$, which are expressed as:

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{bmatrix} \tag{3.10}$$

$$\mathbf{R}_2 = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \tag{3.11}$$

$$\mathbf{R}_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

Additionally there are three translations t_1, t_2, t_3 along these three coordinate axes, which can be simply expressed as a 3×1 vector $\mathbf{T} = [t_1, t_2, t_3]^T$. The above listed rotations and translations can be summarised into a 3×4 matrix, i.e. the camera extrinsic parameter matrix $\begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix}$ [128, 132, 133], as:

$$\begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \tag{3.13}$$

Finally, the camera intrinsic and extrinsic matrices can be incorporated into the projection relation matrix $\mathbf{P}$, which is upgraded according to:

$$\mathbf{P} = \mathbf{K}_{\text{cam}} \begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \tag{3.14}$$

3.2.2 Perspective Projection

In the pinhole camera model, the geometric mapping that projects a 3D scene to several 2D images is called a perspective projection, where the 2D images are acquired by using cameras looking at the 3D scene lying on a 2D plane [128]. Without loss of generality, the plane can be assumed to be the plane where $x_3 = 0$, as:

$$s \begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \tag{3.15}$$

In simple notation, as:

$$s\mathbf{Y}_{\text{2D}} = \mathbf{P}'\mathbf{X}_{\text{2D}} \tag{3.16}$$

where s is any scale factor, and $\mathbf{P}'$ is the new projection relation matrix, while $\mathbf{X}_{\text{2D}}$ is the new 2D homogeneous coordinate vector of the 3D scene. Then, two arbitrary

points Q_a and Q_b that are each projected by the two respective cameras of the same 3D scene point P are related according to:

$$\mathbf{P}_a'^{-1} s_a \mathbf{Y}_{2\mathrm{D}_a} = \mathbf{P}_b'^{-1} s_b \mathbf{Y}_{2\mathrm{D}_b} \tag{3.17}$$

This results in the following relation:

$$\begin{aligned} \mathbf{Y}_{2\mathrm{D}_a} &= \frac{s_b}{s_a} \mathbf{P}'_a \mathbf{P}_b'^{-1} \mathbf{Y}_{2\mathrm{D}_b} \\ &= \mathbf{H} \mathbf{Y}_{2\mathrm{D}_b} \end{aligned} \tag{3.18}$$

where $\mathbf{H}$ is a 3×3 matrix, called the homography matrix, and it is defined as:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{3.19}$$

The above homography matrix $\mathbf{H}$ represents the mathematical relationship between the spatial coordinates of arbitrary two images of the same scene. It has 8 degrees of freedom, since it can be scaled so that the last element h_{33} is always unity, as:

$$\mathbf{H} = h_{33} \begin{bmatrix} h_{11}/h_{33} & h_{12}/h_{33} & h_{13}/h_{33} \\ h_{21}/h_{33} & h_{22}/h_{33} & h_{23}/h_{33} \\ h_{31}/h_{33} & h_{32}/h_{33} & 1 \end{bmatrix} \tag{3.20}$$

The basic set of five 2D planar transformations, i.e. translation, Euclidean, similarity, affine and projective, are listed below (arranged from the simplest one to the most complex one, and they form a nested set, i.e. each simpler one is a subset of the more complex one below it [135]). Examples of the same image each under the translation, Euclidean, similarity, affine and projective transformation distortions are shown in Figure 3.2b, Figure 3.2c, Figure 3.2d, Figure 3.2e and Figure 3.2f, respectively.

Translation 2D Translation is the simplest 2D transformation model, where it preserves all the original properties. The corresponding 2D spatial relationship functions f_x and f_y can be written as:

$$x' = x + t_x$$

and

$$y' = y + t_y \tag{3.21}$$

where t_x and t_y are the translations along the x and y directions, respectively. The corresponding homography matrix $\mathbf{H}$ has 2 degrees of freedom, and it is formulated as:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.22}$$

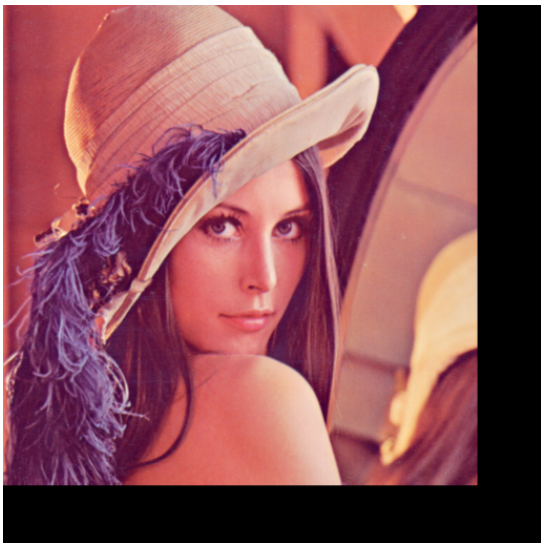
Euclidean This transformation consists of a rotation and a translation, and it is called the Euclidean transformation or the rigid transformation because of the property that preserves all Euclidean distances [18,135]. It also preserves the straightness of lines and all nonzero angles between straight lines. The corresponding 2D spatial relationship functions f_x and f_y can be written as:

$$x' = (x\cos\theta - y\sin\theta) + t_x$$

and

$$y' = (x\sin\theta + y\cos\theta) + t_y \tag{3.23}$$

(a) Original



(b) Translation distorted

(c) Euclidean distorted

(d) Similarity distorted

(e) Affine distorted (f) Projective distorted

Figure 3.2: An example of the same image under different distortions: translation, Euclidean, similarity, affine, and projective.

where θ is a rotation angle about the origin. The corresponding homography matrix $\mathbf{H}$ has 3 degrees of freedom, and it is formulated as:

$$\mathbf{H} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.24}$$

Similarity Compared to the Euclidean transformation above, the similarity allows the addition of a uniform scale change (i.e. no change in the aspect ratio). It is also known as shape preserving mapping, because it preserves straightness of lines and angles between them [18, 135]. The corresponding 2D spatial relationship functions f_x and f_y can be written as:

$$x' = s(x\cos\theta - y\sin\theta) + t_x$$

and

$$y' = s(x\sin\theta + y\cos\theta) + t_y \tag{3.25}$$

where s is a scale change. The corresponding homography matrix $\mathbf{H}$ has 4 degrees of freedom, and it is formulated as:

$$\mathbf{H} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.26}$$

Affine This is more general than the similarity transformation above. It consists of a nonuniform rather than uniform scale change (i.e. there may be a change in the

aspect ratio) and the addition of a shear mapping. The nonuniform scale change can be written as:

$$\begin{aligned} x' &= s_x x \\ y' &= s_y y \end{aligned} \tag{3.27}$$

where s_x and s_y are the scale changes along the respective x and y directions, and the shear mapping can be written as:

$$\begin{aligned} x' &= x + ay \\ y' &= bx + y \end{aligned} \tag{3.28}$$

where a and b are the shear factors, which cause pixel distortions along the x and y directions, respectively. Therefore, 2D affine transformation can tolerate more complex distortions, while maintaining straightness and parallelism of the lines [18, 135]. The corresponding 2D spatial relationship functions f_x and f_y can be written as:

$$x' = a_{11}x + a_{12}y + t_x$$

and

$$y' = a_{21}x + a_{22}y + t_y \tag{3.29}$$

where a_{11}, a_{12}, a_{21} and a_{22} are incorporated into the changes of a rotation, a nonuniform scale and a shear mapping. The corresponding homography matrix $\mathbf{H}$ has 6 degrees of freedom, and it is formulated as:

$$\mathbf{H} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.30}$$

Projective This transformation is the most general one, also known as perspective transformation or homograph, where it employs the homography matrix $\mathbf{H}$ directly. It is a nonlinear transformation, and it only preserves straight lines being straight [18,135]. The corresponding 2D spatial relationship functions f_x and f_y can be written as:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + 1}$$

and

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1} \tag{3.31}$$

The corresponding homography matrix $\mathbf{H}$ has 8 degrees of freedom, and it is formulated as:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \tag{3.32}$$

3.3 Area Based Techniques

Area based techniques for image registration, also known as correlation-like methods or template matching [139] do not really follow the standard registration steps. Firstly, rather than attempt to detect any salient features, they often deal directly with image intensity values i.e. they consider areas of the image (e.g. entire images or windows of predefined size) as features [140–142]. Secondly, they often merge the second and third registration steps (matching features and estimating transformation model) i.e. once feature correspondences are obtained, the parameters of the transformation model are estimated [18]. They share the same idea: first establish a suitable similarity measure function, then shift or warp the images relative to each other, so that the resultant error metric is minimised. The most obvious approach is to do an exhaustive search, i.e. to try all possible transformations and then select the optimal one. Although it is very easy to implement in practice, but it generates a very high computational load. To address the problem, the Fourier transform is often employed to speed up the computation.

Cross Correlation

A classic representative of the area based methods is to perform correlation to assess differences of image intensity values, i.e. to maximise the cross correlation (CC) of the two aligned images [135, 143]. CC itself is not a registration method, but a basic statistical measure of the degree of similarity between two images [17]. In general, it is applicable for images that are misaligned by small translation transformations, because the CC of two images reaches its peak when they are exactly matched [144]. Thus, by computing CC over all possible translations, it is possible to find the exact translation that causes the misalignment between the two images. The CC of the reference image $I_R(x, y)$ and the sensed image $I_S(x, y)$ at a translation of (t_x, t_y) is defined as:

$$CC(t_x, t_y) = \sum_x \sum_y I_R(x, y) I_S(x + t_x, y + t_y) \tag{3.33}$$

However, the CC measure may be unduly influenced by local image intensity value [17]. For example, if there exits a patch in the sensed image $I_S(x, y)$ that possesses a very high intensity value, then the maximum CC may lie in that area. Because of this, CC is normally normalised. The normalised cross correlation (NCC) of the reference image $I_R(x, y)$ and the sensed image $I_S(x, y)$ at a translation of (t_x, t_y) is computed as:

$$NCC(t_x, t_y) = \frac{\sum_x \sum_y (I_R(x, y) - \mu_R)(I_S(x + t_x, y + t_y) - \mu_S)}{\sqrt{\sum_x \sum_y (I_R(x, y) - \mu_R)^2 (I_S(x + t_x, y + t_y) - \mu_S)^2}} \tag{3.34}$$

where μ_R and μ_S are the means of the corresponding images, and they are computed according to:

$$\mu_R = \frac{1}{N} \sum_x \sum_y I_R(x, y) \tag{3.35}$$

$$\mu_S = \frac{1}{N} \sum_x \sum_y I_S(x + t_x, y + t_y) \tag{3.36}$$

where N is the total number of pixels in the image.

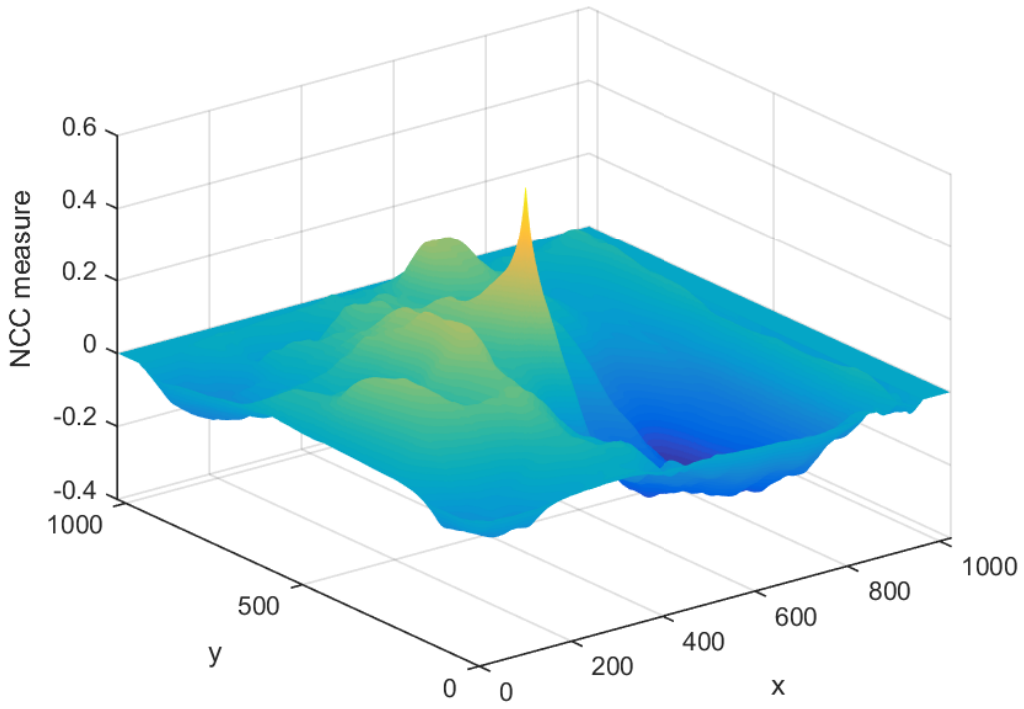
Although the straightforward solution is to do a full search for the maximum value over all possible range of shifts, the corresponding computational complexity becomes significant when the images to be tested are large. To address this problem, the Fourier transform is used. This is based on the Cross-Correlation Theorem, which states that the CC of two images in the spatial domain corresponds to the product of them in the Fourier domain, where one of them has been complex conjugated [17, 145]. Then the CC of the images can be computed more efficiently with the use of the Fast Fourier Transform (FFT) [145]. However, FFT is only applicable for the CC, not for the NCC [17].

Unfortunately, the CC measure is only reliable when the translation shift between two images is fairly small compared to the image size [135]. To address the problem where there is only a small amount of overlap between two images, a windowed variant of CC is normally preferred. Instead of using the entire image, a window of predefined size is employed to obtain the CC measure [140–142]. The windowed cross correlation (WCC) of the reference image $I_R(x, y)$ and the sensed image $I_S(x, y)$ at a translation of (t_x, t_y) is defined as:

$$WCC(t_x, t_y) = \sum_x \sum_y w_R(x, y) I_R(x, y) w_S(x + t_x, y + t_y) I_S(x + t_x, y + t_y) \tag{3.37}$$

where $w_R(\cdot, \cdot)$ and $w_S(\cdot, \cdot)$ are the windowing functions, and they are each padded with are from corresponding images to obtain valid image templates.

Originally, the CC measure was only applicable for aligning translation distorted images, regardless of whether there existed a slight change in rotation and/or scale. To make it tolerate more geometrically deformed images, several generalised variants of the CC have been proposed [142, 146, 147]. They compute the CC of the reference and sensed images, where the sensed image is transformed by each assumed geometric transformation. In other words, the sensed image is transformed for each possible change of interest in translation, rotation, and scale [17]. It results in being able to deal with even more complicated geometric distortions than translation [18]. However, as the number of allowable transformation instances grows, the computational cost easily becomes un-manageable [17].



(a) the NCC of the reference and sensed images

(b) registered result (c)

Figure 3.3: An example of registering a translation distorted image by NCC - the reference and sensed images are shown in Figure 3.2a and 3.2b, respectively: (c) the difference between the reference and registered images.

Sequential Similarity Detection Algorithm Sequential Similarity Detection Algorithm (SSDA) [141] is a method that operates in a similar way to CC. It employs a computationally simpler distance measure than CC, i.e. the accumulated sum of

absolute differences (i.e. 1-norm distance) of the image intensity values, as:

$$SSDA(t_x, t_y) = \sum_x \sum_y \|I_R(x, y) - I_S(x + t_x, y + t_y)\|_1 \tag{3.38}$$

Additionally, it introduces a sequential search strategy to reduce the computational complexity. For any translation candidate of the sensed images, if its corresponding accumulated sum exceeds the predefined threshold, it is rejected and the next translation candidate is tested. In general, SSDA works faster but less accurately, compared to CC [17, 18].

CC, SSDA, and other similar methods often suffer from two major problems. One is the flatness of the similarity measure when searching for the maxima. This problem is mainly caused by the images that possesses the property of self-similarity, which indicates there are similar patches at different pixel locations within the image. To address this problem, pre-processing or the use of the edge or vector correlation is often employed to sharpen the maximum values [142, 148, 149]. The other is high computational complexity [18].

Phase Correlation

To address the problem of high computational complexity and being sensitive to correlated or frequency dependent noise in the correlation-like methods above, images are often operated on in the frequency domain [17, 18]. The phase correlation method was proposed to estimate the translation misalignment between two images, based on the shift property of the Fourier transform [17, 18]. It attempts to search the location of the peak in the inverse of the normalised cross-power spectrum, which is computed based from the tested images. Assuming that there is only a translation displacement between the reference and sensed images I_R and I_S, as:

$$I_R(x, y) = I_S(x + t_x, y + t_y) \tag{3.39}$$

Then, the corresponding 2D Fourier transforms of both sides are related by:

$$\mathcal{I}_R(u, v) = \mathcal{I}_S(u, v) e^{2\pi j(ut_x + vt_y)} \tag{3.40}$$

where $\mathcal{I}_R(u, v) = \mathcal{F}\{I_R(x, y)\}$ and $\mathcal{I}_S(u, v) = \mathcal{F}\{I_S(x, y)\}$ are the 2D Fourier transforms of $I_R(x, y)$ and $I_S(x, y)$ respectively, and (u, v) corresponds to (x, y) in the frequency domain. A shift in the spatial domain is reflected by a phase change in the frequency domain. The normalised cross-power spectrum can easily be obtained through:

$$\begin{aligned} \mathcal{PC}(u, v) &= \frac{\mathcal{I}_S(u, v)\mathcal{I}_R^*(u, v)}{\left\|\mathcal{I}_S(u, v)\mathcal{I}_R^*(u, v)\right\|_2} \\ &= e^{-2\pi j(ut_x + vt_y)} \end{aligned} \tag{3.41}$$

Finally, the phase of the normalised cross-power spectrum can be represented in the spatial domain by taking the inverse 2D Fourier transform of $\mathcal{PC}(u, v)$, as:

$$PC(x, y) = \delta(x - t_x, y - t_y) \tag{3.42}$$

where $PC(x, y)$ is an impulse function, i.e., a single spike is located at the location (t_x, t_y), and approximately zero anywhere else. The final searching process for the peak is quite straightforward.

This achieves a significant reduction in computational complexity, when the images to be registered are large [17,18]. Also, it shows excellent robustness against correlated and frequency dependent noise, and even changes in intensity that are caused by obtaining images under different conditions of illumination [17, 18]. This is because changes in illumination are usually slowly varying, thus they are concentrated at low frequencies [17]. While PC is good at dealing with the images that are corrupted by noise in a narrow frequency band, e.g. low frequency noise [17, 135], it is not applicable for the images that possess significant white noise that is spread across all frequencies [17], or have very low signal-to-noise ratio at some frequencies [135].

Extension to Similarity Distortion Originally, PC was proposed for registering images misaligned by translations. Further work [150–152] has demonstrated that, under certain limited conditions, PC can be improved to tolerate more complicated geometric distortions up to those caused by a similarity transformation. The three unknown parameters (i.e. rotation, scale, and translational shift) can be estimated using the following steps: First, both images are converted to the frequency domain by the 2D Fourier transform. Then, only the magnitudes of the converted images are kept, because they are insensitive to translations in the spatial domain. Next, after resampling both magnitude images into the log-polar coordinates, the parameters for rotation and scale can be estimated by a standard PC method. Finally, one of the original images can be de-rotated and scaled, so that the last parameter for translational shift can be estimated by another regular PC method.

Assume that the reference and sensed images I_R and I_S are related by a similarity transformation, i.e. a translation (t_x, t_y), a rotation θ and a uniform scale change s, as:

$$I_R(x, y) = I_S(s(x\cos\theta - y\sin\theta) + t_x, s(x\sin\theta + y\cos\theta) + t_y) \tag{3.43}$$

Then, the 2D Fourier transforms of both sides are related by:

$$\begin{aligned}\mathcal{I}_R(u, v) =& \frac{1}{s^2}\mathcal{I}_S\left(\frac{u\cos\theta - v\sin\theta}{s}, \frac{u\sin\theta + v\cos\theta}{s}\right) \\ &\times e^{2\pi j\left(\frac{u\cos\theta - v\sin\theta}{s}t_x + \frac{u\sin\theta + v\cos\theta}{s}t_y\right)}\end{aligned} \tag{3.44}$$

Next, the magnitude images can be easily obtained as:

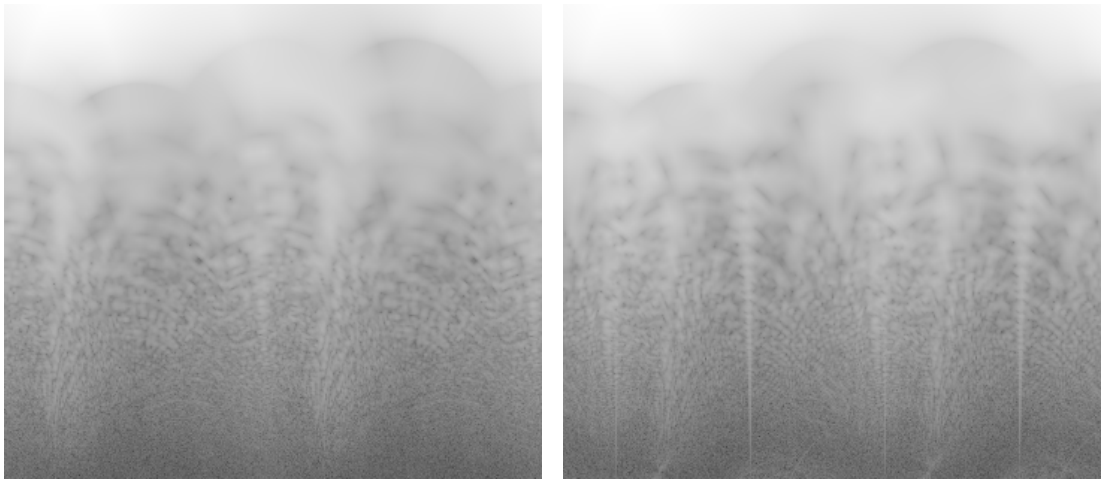
$$\mathcal{M}_R(u, v) = \frac{1}{s^2}\mathcal{M}_T\left(\frac{u\cos\theta - v\sin\theta}{s}, \frac{u\sin\theta + v\cos\theta}{s}\right) \tag{3.45}$$

where $\mathcal{M}(u,v) = \|\mathcal{I}(u,v)\|_2$ is the magnitude image. Then, both magnitude images are converted into polar coordinates, with $u = r\cos\varphi$ and $u = r\sin\varphi$, as:

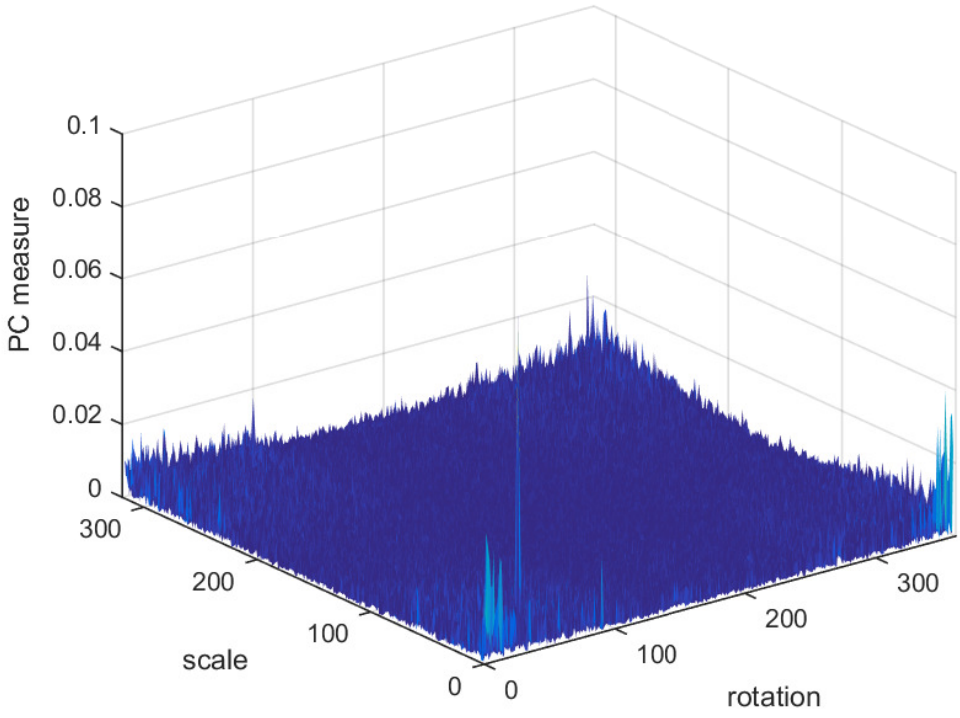
$$\mathcal{M}_R(r\cos\varphi, r\sin\varphi) = \frac{1}{s^2}\mathcal{M}_T\left(\frac{r\cos(\varphi+\theta)}{s}, \frac{r\sin(\varphi+\theta)}{s}\right) \tag{3.46}$$

and (3.46) above can be reformulated as:

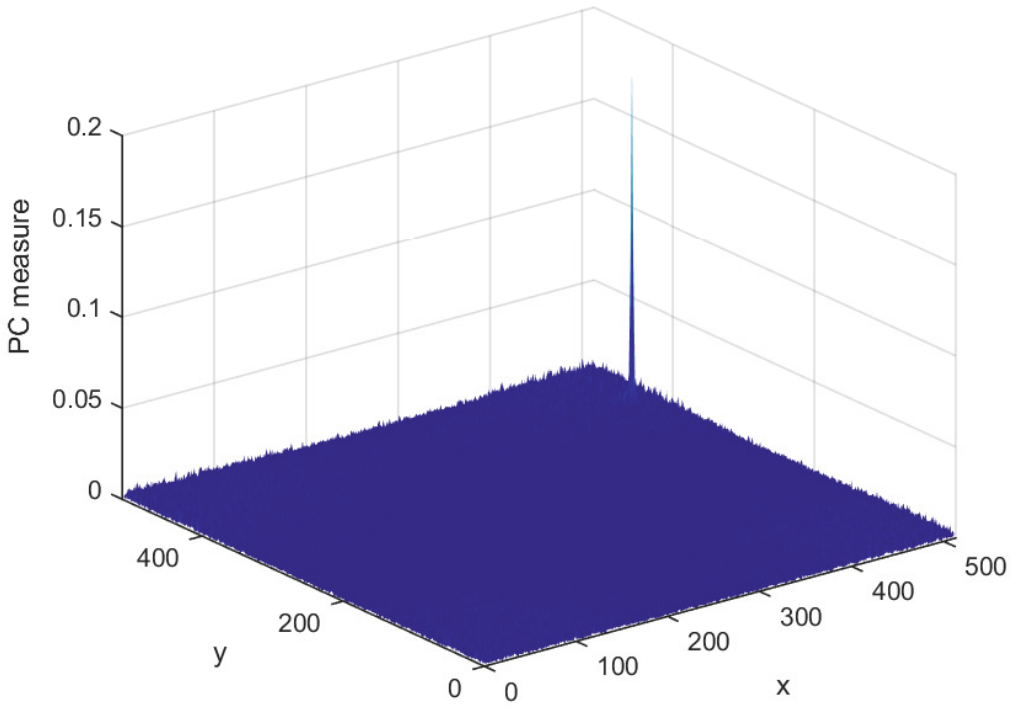
$$\mathcal{M}_R(r,\varphi) = \frac{1}{s^2}\mathcal{M}_T\left(\frac{r}{s}, \varphi+\theta\right) \tag{3.47}$$



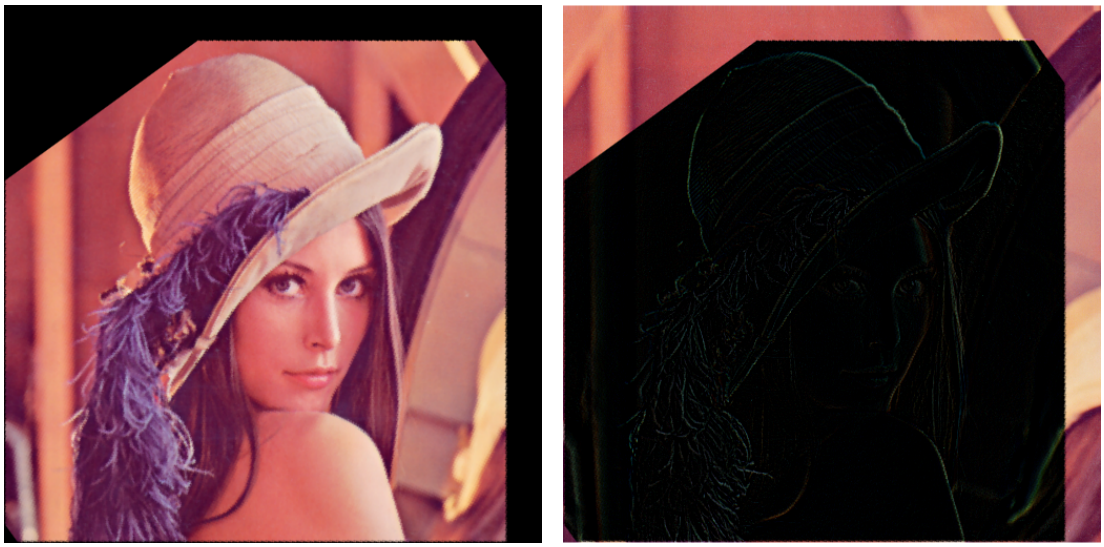
(a) (b)



(c) the PC for the estimate of rotation and scale



(d) the PC for the estimate of translation



(e) registered result (f)

Figure 3.4: An example of registering a similarity distorted image by PC - the reference and sensed images are shown in Figure 3.2a and 3.2d, respectively: (a) and (b) the spectrum magnitudes of the reference and sensed images in the log-polar coordinates, respectively (f) the difference between the reference and registered images.

After taking the logarithm of both sides, both images can be further converted into

log-polar coordinates,

$$\mathcal{M}_R(\log r, \varphi) = \frac{1}{s^2}\mathcal{M}_T(\log r - \log s, \varphi + \theta) \tag{3.48}$$

Compared the above equation to eq. (3.39), it can be found that they share the same formulation except that eq. (3.48) has a scaling factor $1/s^2$ on its right side. Then, regardless of the scaling factor $1/s^2$, both parameters θ and s for respective rotation and scale can be easily estimated, with the use of a standard PC method.

3.3.1 Sub-pixel Precision

The above described methods can only estimate transitional shift to the nearest neighbouring pixel location, i.e. an integer rather than a fractional value. If there is a demand for higher accuracy of the registration results, sub-pixel precision becomes necessary. A possible approach is to evaluate several discrete values around the pixel location, where it holds the best registration result after the estimation process [153]. A more commonly used approach is to employ gradient descent to get the optimal values, with the use of a Taylor series expansion of the function of the image intensity value [154]. Assuming that the reference and transformed sensed images $I_R(x, y)$ and $I_S(x + t_x, y + t_y)$ will be exactly matched with a sub-pixel displacement $(\Delta t_x, \Delta t_y)$, as:

$$\sum_x \sum_y \left(I_S(x + t_x + \Delta t_x, y + t_y + \Delta t_y) - I_R(x, y)\right) = 0 \tag{3.49}$$

The solution of $(\Delta t_x, \Delta t_y)$ is found by minimising the following function:

$$\begin{aligned} SP &= \sum_x \sum_y \|I_S(x + t_x + \Delta t_x, y + t_y + \Delta t_y) - I_R(x, y)\|_2 \\ &\approx \sum_x \sum_y \|I_S(x + t_x, y + t_y) + \mathbf{J}_S(x + t_x, y + t_y)(\Delta t_x, \Delta t_y) - I_R(x, y)\|_2 \\ &= \sum_x \sum_y \|\mathbf{J}_S(x + t_x, y + t_y)(\Delta t_x, \Delta t_y) - I_{R-S}(x, y)\|_2 \end{aligned} \tag{3.50}$$

where $\mathbf{J}_S(x + t_x, y + t_y)$ is the gradient of the sensed image I_S at the pixel location $(x + t_x, y + t_y)$, as:

$$\begin{aligned} \mathbf{J}_S(x + t_x, y + t_y) &= \nabla I_S(x + t_x, y + t_y) \\ &= \left(\frac{\partial I_S}{\partial x}, \frac{\partial I_S}{\partial y}\right)(x + t_x, y + t_y) \end{aligned} \tag{3.51}$$

and $I_{R-S}(x, y) = I_R(x, y) - I_S(x + t_x, y + t_y)$ is the current error of the intensity value. The above problem can be minimised by the following steps [155]:

$$\nabla SP = 2\sum_x \sum_y \mathbf{J}_S^T(x + t_x, y + t_y)(\mathbf{J}_S(x + t_x, y + t_y)(t_x, t_y) - I_{R-S}(x, y)) \tag{3.52}$$

Then, setting $\nabla SP = 0$:

$$\mathbf{H}_{GN}(t_x, t_y) = \mathbf{r}_{gw} \tag{3.53}$$

where $\mathbf{H}_{GN}$ and $\mathbf{r}_{gw}$ are called the Gaussian-Newton approximate Hessian and gradient-weighted residual vector, respectively. They are defined as [155]:

$$\mathbf{H}_{GN} = \sum_x \sum_y \mathbf{J}_S^T(x + t_x, y + t_y)\mathbf{J}_S(x + t_x, y + t_y) \tag{3.54}$$

and

$$\mathbf{r}_{gw} = \sum_x \sum_y I_{R-S}(x, y)\mathbf{J}_S^T(x + t_x, y + t_y) \tag{3.55}$$

If there is a concern about computational loads, the gradients in the sensed image can be approximated by the gradients in the reference image as:

$$\mathbf{J}_S(x + t_x, y + t_y) \approx \mathbf{J}_R(x, y) \tag{3.56}$$

This is based on the fact that near correct alignment, the reference and the displaced sensed images should have very similar intensity values. It has the advantage of allowing the pre-computation of these gradients, which can result in a significant improvement in efficiency [156, 157].

The effectiveness of the method above is dependent on the accuracy of the Taylor series expansion. It may be necessary to have several iterations when the initial points are far away from the exact displacement, e.g. 1-2 pixels. There is a trade-off between the quality of the correctness and the number of iterations. It is common to choose a stopping criterion based on the magnitude of the displacement correction, i.e. $\|(\Delta t_x, \Delta t_y)\|_2$: when it is below a certain threshold (e.g. 0.1 pixel) the iteration stops [135].

3.4 Feature Based Techniques

While the area based registration techniques work directly to minimise dissimilarities between intensity values of the corresponding pixel locations, a different group of methods work by extracting salient structure features from images and then matching them to each other. The extracted features should be distinct, efficiently detectable, and expected to be stable in both images [18]. The potential candidates can be significant regions, lines or even points [18]. Compared to area based methods, the feature based approaches have the advantage of being more robust against a movement in the scene [18, 135], which results in making them suitable for situations when large changes (e.g. changes in illumination) are expected in the images or the images are obtained from different sensors [18, 122]. The feature based techniques for image registration usually follow the standard steps: i.e. first detecting distinctive features in each image, then matching these features to establish correspondences, and finally estimating the geometric transformation models between the images.

3.4.1 Detecting Features

Detected features are matched to get a set of correspondences between two images, and the matched pairs are then used to estimate the transformation model so that the two images can be geometrically aligned. The process of detecting features therefore becomes the most crucial stage. The similarity between the same features detected from different images should be sufficiently high, regardless of existence of image geometry, presence of additive noise, and even changes in the scene. As a result, the detected features possess the property of being distinct, somehow stable and efficiently detectable. In general, features should be detected so that they are appropriate for the encountered tasks. Mainly, there are three types of features that can be detected in an image: region-like, line-like, and point-like.

For region-like features, the potential candidates are the projections of closed boundary regions of an appropriate size, whose contrast is general high [158,159]. For example, water regions (e.g. lakes) [160, 161], or buildings [162] in aerial imagery. The corresponding centres of gravity (i.e. the average location of the weight of an object [163]) are often used to represent these features, because they are recognisable with respect to changes in rotation, scaling, skewing, random noise, and grey-level (i.e. intensity) [18]. Usually, these kinds of features can be identified by a variant of segmentation techniques [159,164], such as k-means clustering [165], watershed transformation [166], and Otsu's method [167].

The reasonable candidates for line-like features are the representations of general line segments [162, 168, 169], e.g. object contours [170–172] or roads [173]. This type of feature is suitable for detecting elongated anatomical structures [174] and geological elements, they are mainly used in registration problems for medical and satellite images [18]. They are often expressed by the middle or end points of lines, so that correspondences can be established [18]. Standard approaches to edge detection are usually employed to identify line-like features e.g. the first-order Canny edge detector [175], or second-order detectors based on the Laplacian of Gaussian [176].

Finally, for point-like features, there are many potential candidates: identification of line intersections [177, 178], points possessing high variance [179], discontinuities of local curvature [180, 181], inflection points of curves [182, 183], and local extrema of wavelet transforms [184, 185]. Most techniques employed for detecting point-like features are based on the idea of defining corners [162, 186–190], which are usually points of high curvature on the region boundaries [18]. This is because corners are invariant to geometric transformations of images, and they are easily perceived by human observers [18].

For simpler processing in the next stage, most of the existing detection techniques use a representative point (e.g. the centre), called a control point, to represent the location of a detected feature. More recently, the study of the methods for feature

detection and descriptor assignment (i.e. constructing a descriptor to represent all of the details of a detected feature) [122, 191–199] continue to be very active [135]. The features detected here are corner-like features [200] (also known as keypoints, or interest points), which can be matched with high accuracy. They are more invariant to changes in scale and affine transformation compared to the region and line-like features, which makes them suited to match images that have different scale and/or aspects [135]. To detect these features, many techniques based on utilising the image Hessian matrix [201] have been proposed [123, 193, 194, 202–206]. To achieve scale invariance, reasonable candidates can be obtained by finding the local maxima in the scale-space of Difference of Gaussian (DoG) [191, 192, 207]. The scale-space of DoG is an image pyramid, with a subsampling factor between two neighbouring image levels less than 2. To make a keypoint distinguished from others, it should be localised with a high accuracy, and its corresponding descriptor should be constructed to be invariant under assumed image degradation and be sufficiently discriminable.

Scale Invariant Feature Transform

Scale Invariant Feature Transform (SIFT) [191, 192] is a method that detects invariant keypoints based on a high level consideration of the image structure. The detected keypoints are invariant to location, scale and rotation, and robust to affine transformations as well as changes in illumination [191, 192]. Additionally, for each keypoint (i.e. a circular image patch with an orientation), SIFT assigns an appropriate descriptor vector of dimension 128, along with several corresponding parameters: the centre location, the radius, and the orientation of the keypoint.

Scale-space Extrema Detection Scale-space is a theory that represents an image as a family of blurred images with a variable scale, so that image structures at different scales can be detected [208, 209]. To achieve scale-space representation, it has been shown that under certain assumptions, the Gaussian kernel is a reasonable approach [209, 210]. For an arbitrary image $I(x, y)$, its corresponding scale-space representations $L_s(x, y, \sigma)$ at the scale of σ are computed by:

$$L_s(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3.57}$$

where $G(x, y, \sigma)$ is a Gaussian kernel of the scale σ as [211, 212]:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{3.58}$$

and $*$ is the convolution operator.

It has been demonstrated by the detailed experimental comparisons conducted in [213], that the extremum of the scale-normalised Laplacian of Gaussian $\sigma^2\nabla^2 G$ corresponds to the most stable image features, compared to a variety of other possible

approaches (e.g. Harris-Affine Regions [214] and Harris-Laplace [215]). Additionally, the work described in [209] shows that the DoG function approximately equals the scale-normalised Laplacian of Gaussian $\sigma^2\nabla^2 G$. Thus, to effectively detect the locations of stable keypoints, it is reasonable to search the extrema in the DoG function, which can be computed from the difference between two consecutive scale-space representations separated by a multiplicative factor of constant k, as:

$$D_g(x, y, \sigma) = L_s(x, y, k\sigma) - L_s(x, y, \sigma) \tag{3.59}$$

To detect the local extremum of $D_g(x, y, \sigma)$, an efficient way has been suggested [213]. First, several octaves of scale-space images (i.e. a image pyramid) are generated based on the original image, as shown in the left column of Figure 3.5. The image size of each octave is half, compared to that of its previous one (i.e. doubling of the scale σ). Within each octave, the blurred images are separated by a constant factor k. To divide each octave into an integer number s of intervals, k should be chosen so that $k = 2^{1/s}$. Additionally, to search a complete octave for final extrema detection, there should exist $s + 3$ blurred images for each octave. In [213], s is recommended to be set to be $s = 2$. Then, within each octave, neighbouring blurred images are subtracted to generate the corresponding DoG images, as shown in the right column of Figure 3.5. An octave of scale-space images and the corresponding DoG images are generated for the original image shown in Figure 3.2a, as shown in Figure 3.6 and 3.7, respectively. Finally, the intensity value of each pixel location is compared to that of its 26 neighbours in 3×3 patches (i.e. 8 in the current scale image as well as 9 each in the previous and following ones), as shown in Figure 3.8. In [213], for the scale σ, it is set to be $\sigma = 1.6$. To avoid discarding the highest spatial frequencies when smoothing images before detecting extrema, the image can be expanded to make full use of its input. In [213], it is recommended that a bilinear interpolation technique is applied to double the size of input image, before creating the image pyramid.

Keypoint Localisation Some of the detected candidates for keypoints need to be rejected, because they have low contrast (i.e. are sensitive to noise) and/or are poorly localised on an edge. Additionally, because the actual extrema may exist at sub-pixel locations, it is reasonable to localise them precisely by employing a Taylor expansion of the DoG image as [216]:

$$D_g(\mathbf{x}) = D_g + \frac{\partial D_g^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 D_g}{\partial \mathbf{x}^2}\mathbf{x} \tag{3.60}$$

where the employed Taylor expansion has been shifted so that its origin is at the candidate keypoint, $\mathbf{x} = (x, y, \sigma)^T$ is the offset from the candidate keypoint, and the values of D_g as well as its derivatives are evaluated at the candidate keypoint. The actual offset of extremum $\hat{\mathbf{x}}$ can then be determined easily by taking the derivative of

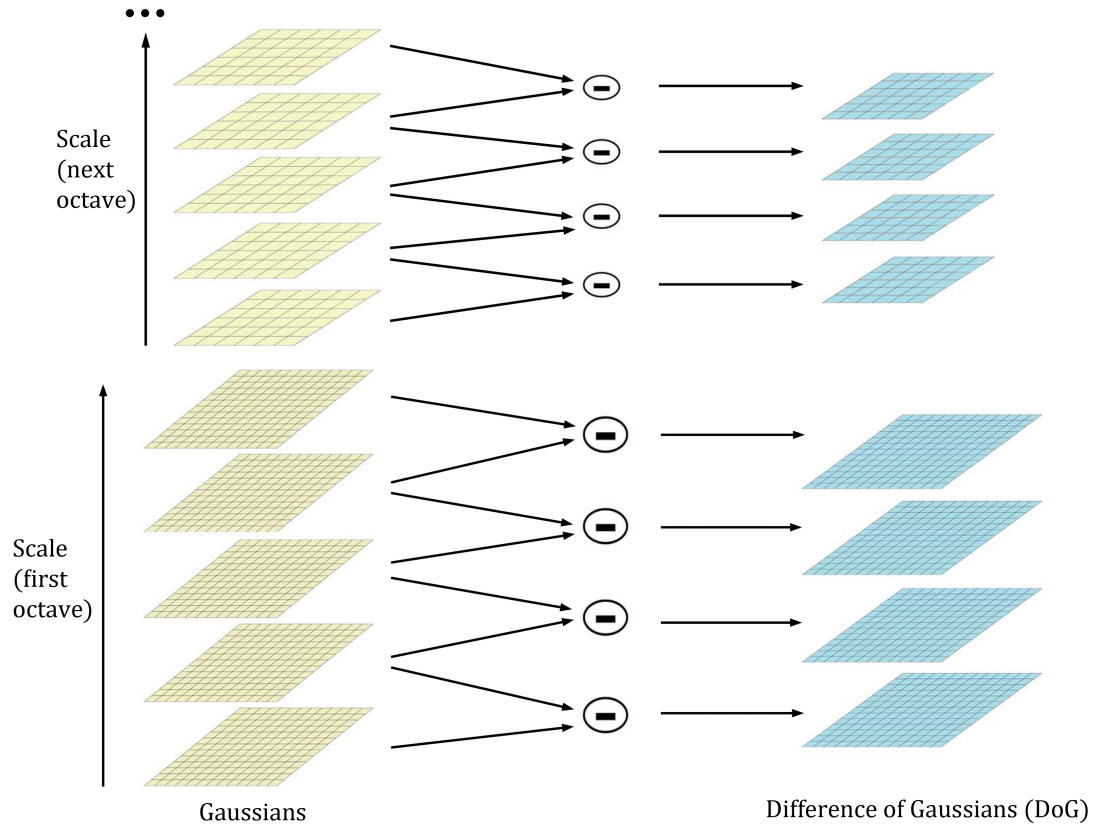


Figure 3.5: An example of constructing DoG images (taken from [192] but redrawn here): within each octave, Gaussian blurred images of the original image are generated so that they are separated by a constant factor k, as shown on the left column. Then, the neighbouring blurred images are subtracted to generate the DoG images, as shown on the right column. After finishing the above process within an octave, the blurred image is downsampling by a factor of 2, and then the process is repeated.

(a) $\sigma = 1.6$ (b) $\sigma = 1.6 \times \sqrt{2}$

(c) $\sigma = 1.6 \times 2$

(d) $\sigma = 1.6 \times 2\sqrt{2}$

(e) $\sigma = 1.6 \times 4$

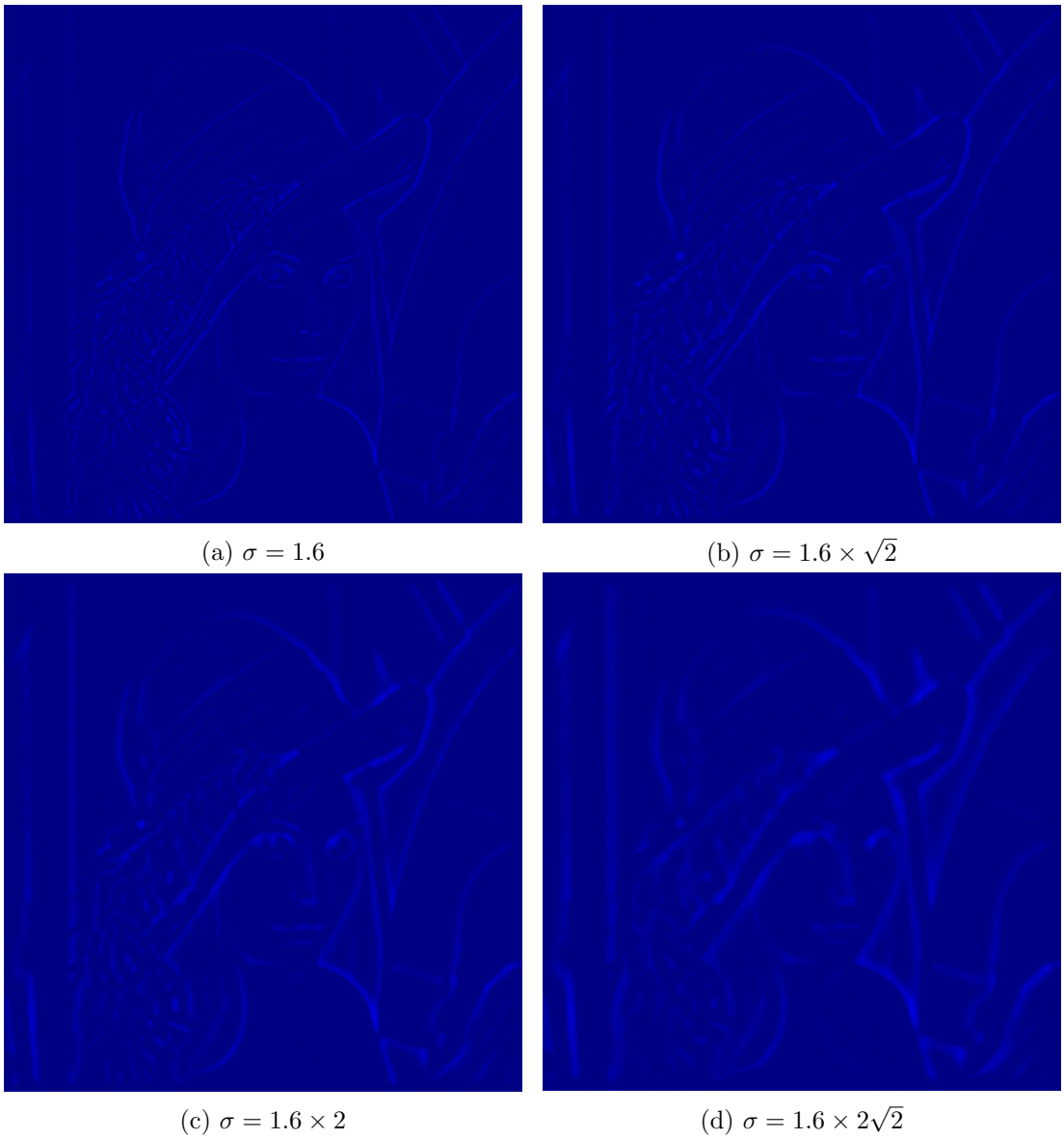
Figure 3.6: An example of scale-space images in an octave for the original image shown in Figure 3.2a.

above function with respect to $\mathbf{x}$ and setting it to zero, as given by:

$$\hat{\mathbf{x}} = -\frac{\partial^2 D_g^{-1}}{\partial \mathbf{x}^2} \frac{\partial D_g}{\partial \mathbf{x}} \tag{3.61}$$

If the obtained offset $\hat{\mathbf{x}}$ is larger than 0.5 in either dimension, then it results in the actual extremum lying closer to a different pixel location. The above process then needs to be repeated for this new pixel location.

Once the final offset $\hat{\mathbf{x}}$ is obtained, the corresponding function value $D(\hat{\mathbf{x}})$ can be obtained. If its value is below a threshold value, then this candidate keypoint is removed, so that candidates with low contrast are excluded. To eliminate a candidate keypoint based on poor localisation, it is noticed that for an edge, there is a large



(a) $\sigma = 1.6$ (b) $\sigma = 1.6 \times \sqrt{2}$

(c) $\sigma = 1.6 \times 2$ (d) $\sigma = 1.6 \times 2\sqrt{2}$

Figure 3.7: An example of DoG images, which corresponds to the scale-space images shown in Figure 3.6: to make them move visible, all images are in the jet colour map.

principal curvature across it but only a small one in the perpendicular direction. If the difference between two curvatures is below the ratio of largest to smallest eigenvector, which are computed from the 2×2 Hessian matrix at the exact location and scale of the candidate keypoint, then this candidate is rejected.

Orientation Assignment The aim is to assign a consistent orientation to each keypoint based on local image properties. One approach to find the orientation is as follows: for each pixel location (x, y) within a region around the keypoint, the magnitude of the gradient mag is computed as:

$$mag(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (3.62)$$

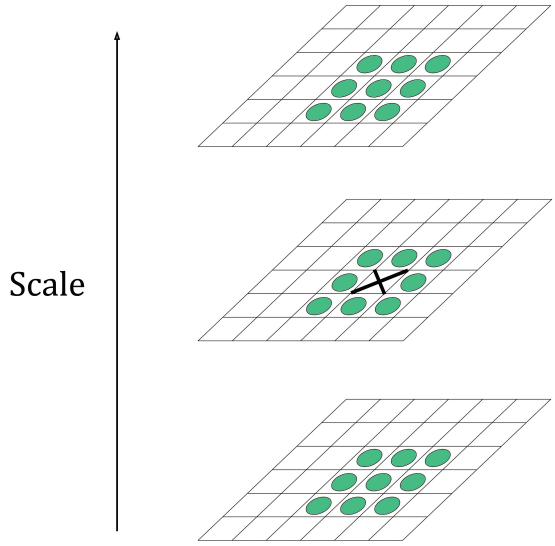


Figure 3.8: An example of detecting extremum in a DoG image (taken from [192] but redrawn here): comparing the intensity value of a pixel location (marked as X) to that of its 26 neighbours at the current and adjacent images (marked as green circles).

where L is the Gaussian blurred image corresponding to the scale of the keypoint, and the orientation of the gradient $\theta(x, y)$ is computed as:

$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \tag{3.63}$$

Next, the computed gradient orientations are used to form an orientation histogram, which has 36 bins covering the whole range of orientations (i.e. each bin covers a range of 10 degrees). Each of them is added to the histogram according to its gradient orientation, and weighted by its gradient magnitude as well as by a Gaussian window whose scale is 1.5 times that of the keypoint. Then, the global peak in the histogram is located. The keypoint is assigned the orientation(s) corresponding to this peak and any other local peak within 80% of the height of the global peak. Multiply assigned orientations will make a significant contribution to the stability of the final matching. Finally, a parabola is fitted to the 3 histogram values closest to each peak to interpolate the peak position with a better accuracy.

Keypoint Descriptor A unique descriptor is assigned to each of the keypoints, so that they can be distinguished from each other. A window of 16×16 pixel locations is centred on the keypoint, then it is further broken into 16 windows whose size is 4×4 pixel locations, as shown in the left of Figure 3.9. Within each 4×4 window, the gradient information (i.e. gradient magnitude and orientation) of all pixel locations is calculated, to form an orientation histogram of 8 bins, which results in a feature vector containing 128 elements, as shown in the right of Figure 3.9. To construct an 8 bin histogram, the gradient information of all 16 pixel locations should be rotated

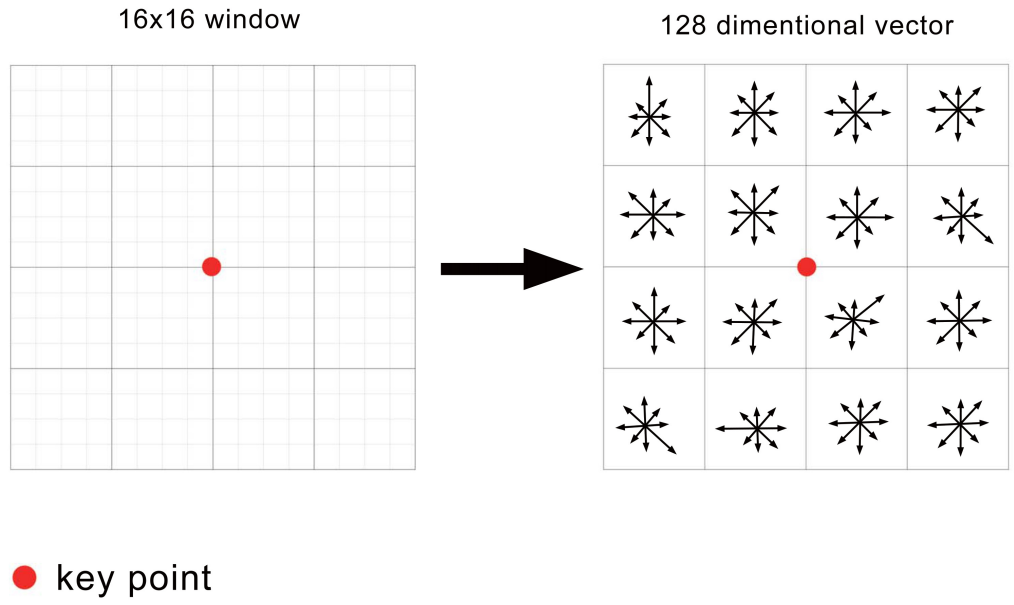


Figure 3.9: An example of constructing a descriptor for a keypoint [192]: within each $4\times$ 4 window, the gradient magnitudes and orientations of all pixel locations are calculated to from an 8 bin histogram.

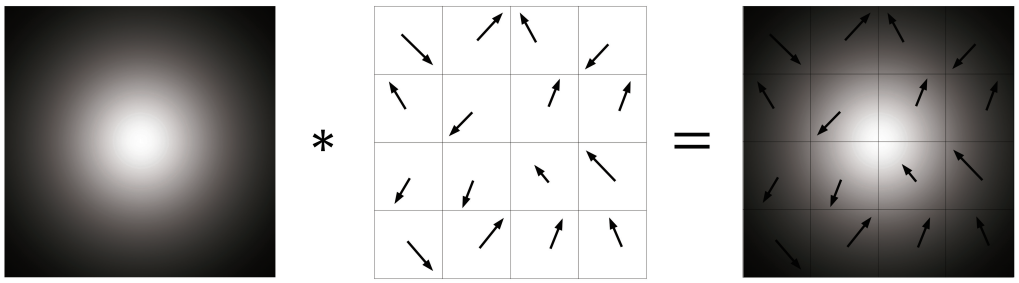


Figure 3.10: An example of constructing an 8 bin histogram [192]: a gradient orientation is added to the bin covering this orientation (i.e. each bin covers 45 degrees), and the amount added to the bin depends on the corresponding gradient magnitude.

to line up with the orientation of the keypoint (i.e. the orientation of the keypoint is subtracted from each orientation) to be invariant to any changes in rotation. Similarly, each pixel location contributes to the histogram according to its gradient magnitude, but the contribution should be weighted by a Gaussian window whose scale is 1.5 times that of the keypoint, as shown in Figure 3.10. Finally, to achieve partial illumination independence, any number (of the 128) greater than a threshold value (e.g. 0.2) is set to the threshold value, and then the resultant feature vector is normalised again.

PCA-SIFT PCA-SIFT [217] is an extension of SIFT. It follows the same processing stages of SIFT, except for the stage that constructs feature descriptors. Instead of weighting the orientation histogram of the blurred patch, it applies PCA to the normalised feature vector, which is obtained from computing the local gradient information of image patch. There are two advantages: employing PCA allows the variations within a image patch to be modelled accurately; and they can be represented using a compact feature vector. One retains the variations that are related to identification and the other discards the distortions that are caused by other effects.

Firstly, at the scale where a keypoint is detected, a region of the size 41×41 is extracted and centred around the keypoint. Then, an eigenspace is computed to express the gradient information of the image patches. For a given patch, the local gradient information in the horizontal and vertical directions are computed, which results in a feature vector of $2 \times 39 \times 39 = 3042$ elements. Finally, the resultant feature vector is projected to the eigenspace computed using PCA, which results in a compact feature vector whose dimension is smaller than that of the original SIFT feature vector.

Speeded Up Robust Features Speeded Up Robust Features (SURF) [193, 194] is a technique that is inspired by SIFT and used to detect features and assign descriptors. It is several times faster than SIFT, while its performance is comparable. It speeds up the computational process mainly because it makes an efficient use of integral images.

In SURF, the Laplacian of Gaussians is approximated by employing a box filter, which results in two advantages: one is that the convolution with the box filter can be obtained easily utilising integral images. The other is that the processing can be done simultaneously for different scales. To determine the values for both scale and location, it relies on the determinant of the Hessian matrix because, for many applications, it is not necessary to achieve invariance to changes in rotation. It offers the option whether to compute the orientation, which results in a further saving in the computational cost. For the descriptor of the keypoint, it uses Haar wavelet responses in the horizontal and vertical directions, which can easily be obtained employing integral images. First a region, whose size is $20s \times 20s$, is centred around the keypoint, where s is the scale at which the keypoint was detected. This is then divided into 4×4 subregions. Within each subregion, the horizontal and vertical Haar wavelet responses d_x and d_y are taken, and they are summed up over the whole subregion to form a vector of dimension 4, as:

$$\mathbf{v}_h = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right) \tag{3.64}$$

This results in a feature descriptor with 64 dimensions. Additionally, it speeds up the matching process. It uses the sign of the trace of the Hessian matrix to indicate whether a keypoint is detected from a bright blob on a dark background or from the reverse situation (i.e. a dark blob on a bright background). Then, in the matching stage, only

the feature descriptors detected from the same situation are compared, which results in faster matching while keeping the same performance.

3.4.2 Matching Features

After features have been detected (e.g. salient points or small blocks), they should be matched so that the locations of corresponding features from different images can be determined. The easiest approach to match all the corresponding features between two images is to compare all the features detected in one image against those detected in the other. However, it should take into account the fact that the physically corresponding features can become dissimilar due to the different image acquisition conditions and/or the different sensor spectral sensitivities, and some features do not have corresponding pairs in the other image. To address this, a variety of more robust methods have been proposed. For example, the chamfer matching algorithm introduced in [218] matches detected line features from different images, so that the generalised distances between them are minimised. The method proposed in [219] is based on relaxation, where it recomputes the figures of merit iteratively for all feature pairs while considering the matching quality of these features pairs, until an optimal result is reached. However, the time taken for these methods makes this kind of approach unsuitable for some practical applications. To devise a more efficient matching scheme, the idea of finding nearest neighbours in the feature space is frequently employed. For example, a technique called "slicing", developed in [220], discards a group of candidates that lie outside a hypercube of a test data sample in the feature space, and then employs a series of 1D binary searches. An algorithm called Best Bin First (BBF) was proposed in [221], where it approximates the k-d tree algorithm [222] (i.e. a space partitioning tree for organising data samples in a k-dimensional space) and returns the nearest neighbours with high probability. It makes the search of bins (i.e. candidates) in the feature space corresponding to the ascending sort of the distances from the test location, by employing a modified search ordering for the k-d tree algorithm. Then, by cutting off the search after a specific number of nearest bins have been explored, an approximate result can be obtained at low computational cost. Unfortunately, there are almost always outliers among the matched results of feature correspondences (i.e. data samples whose distribution fits some model), it is better to use robust model fitting techniques to find a good starting inlier set of feature correspondences (i.e. feature correspondences that fit the estimated transformation model).

RANdom SAmple Consensus

RANdom SAmple Consensus (RANSAC) [223] is an iterative technique for estimating parameters of a mathematical model from a set of data that contains outliers. It is widely applied to model fitting problems, because of its simple implementation and

robustness. It works by assuming that all of the data consist of either inliers that can be explained by a mathematical model with a set of particular parameter values, or outliers that cannot fit that model in any situations. Additionally, it assumes that there exists a procedure for optimally estimating the parameters of the mathematical model from the available data. Although RANSAC is able to estimate the model parameters with high accuracy even when there exists a significant number of outliers, it may not always be able to obtain the optimal solution when the number of computing iterations is insufficient, and it usually achieves a bad performance when the percentage of inliers is less than 50% [223].

RANSAC simply iterates the following steps: generating a hypothesis from randomly selected data and then verifying it on all the data. The random selection of data can result in two advantages: there is no need to employ complex optimisation algorithms nor to take a large amount of memory [224]. It is because that by randomly selecting a minimal subset from the data and then using this selected subset to estimate a mathematical model, the total number of data need to be considered by a model fitting function is significantly reduced. To show how RANSAC iterates the steps, it is applied to find a model for a set of 2D data samples, and two iteration examples are shown in Figure 3.11 and 3.12. In this application, two data samples are randomly selected for model estimation in each iteration, and they can be easily fitted to a linear equation. From Figure 3.11 and 3.12, it can be seen that the estimated model in Figure 3.12 is more appropriate than that in Figure 3.11, where there is a larger number of data samples consist with the model.

Here, for matching detected features, RANSAC starts by randomly selecting a subset of k feature correspondences, which is then used to compute an estimated transformation model. Next, the estimated transformation model is applied to obtain the errors of the full set of feature correspondences, i.e. the differences between mapped estimated locations and the detected feature locations. Then, it determines how many inliers exist, based on whether their corresponding errors are within a predefined tolerance (usually, set to be around 1-3 pixel locations [135]). The above process is repeated S_{sac} times, and the set of feature correspondences which has the largest number of inliers is regarded as the final solution.

The number of iterations S_{sac} must be chosen to be sufficiently large, so that the probability P_{sac} of finding a true corresponding set of inliers is high enough (usually, set to be 0.99 [225]). Let p_{sac} represent the probability that any randomly selected feature correspondence is an inlier. Therefore, the likelihood an iteration that all k randomly selected feature correspondence are inliers is p_{sac}^k. Then, the likelihood that all S_{sac} iterations fail is [225]:

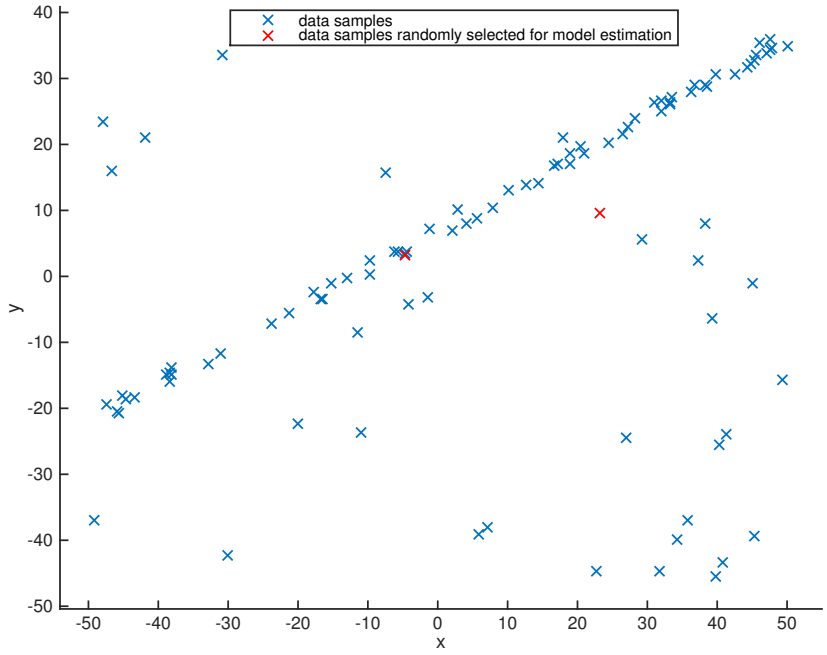
$$1 - P_{sac} = (1 - p_{sac}^k)^{S_{sac}} \tag{3.65}$$

and with some manipulations, the minimum number of iterations required can be ob-

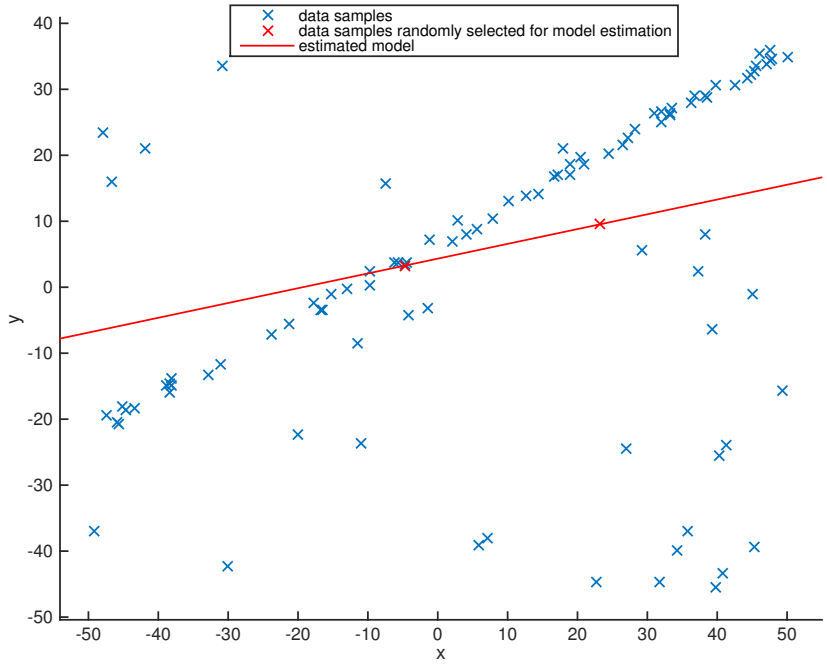
tained as [225]:

$$S_{sac} = \frac{\log(1 - P_{sac})}{\log(1 - p_{sac}^k)} \tag{3.66}$$

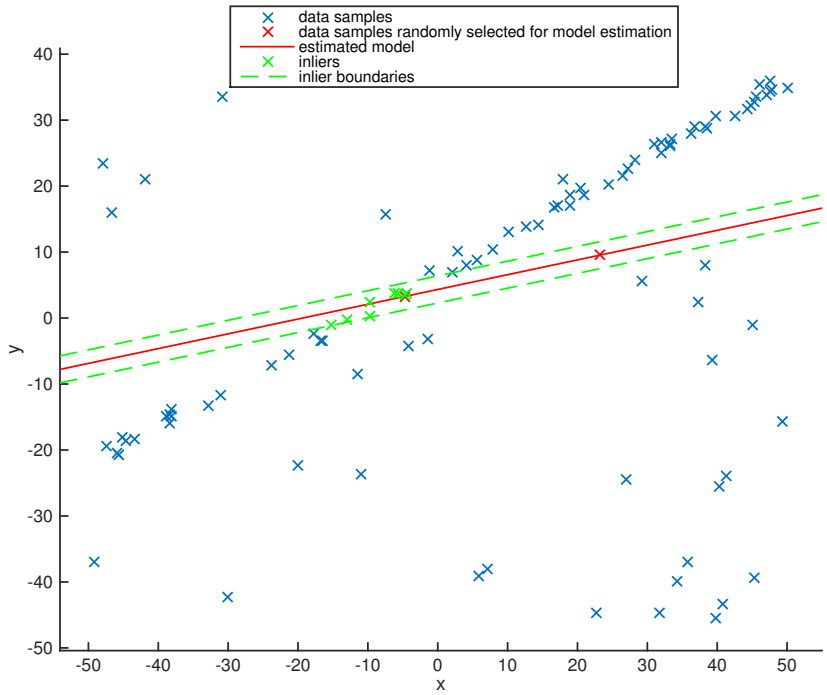
To show how RANSAC matches detected features (e.g. SIFT features) for image regis-



(a) two randomly selected data samples

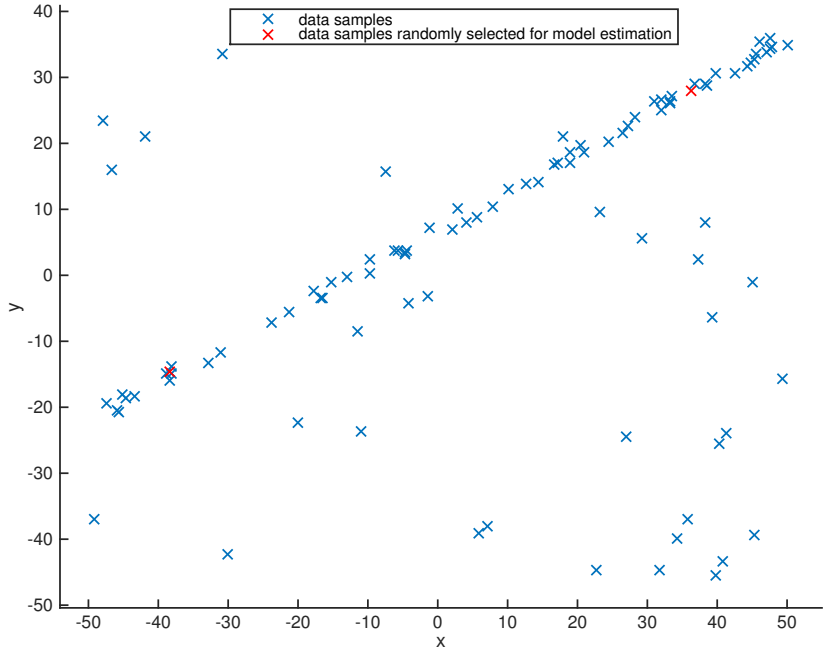


(b) an estimated model based on the two randomly selected data samples

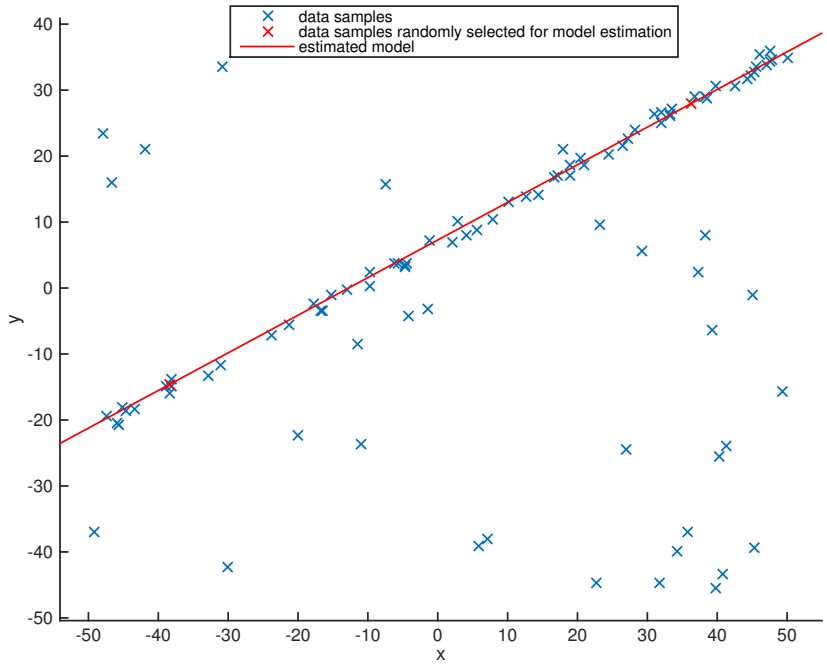


(c) the data samples consistent with the estimated model

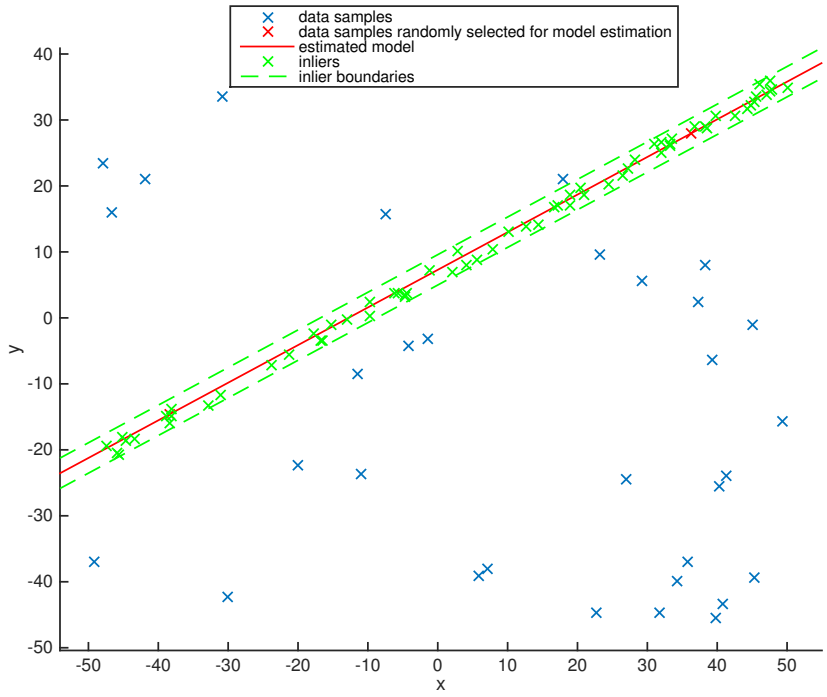
Figure 3.11: A RANSAC iteration example: estimating a model for a set of 2D data samples, based on two randomly selected data samples.



(a) two randomly selected data samples



(b) an estimated model based on the two randomly selected data samples



(c) the data samples consistent with the estimated model

Figure 3.12: Another RANSAC iteration example: estimating a model for a set of 2D data samples, based on two randomly selected data samples.

tration, an example of registering a projective distorted image by SIFT and RANSAC is showed in Figure 3.13.

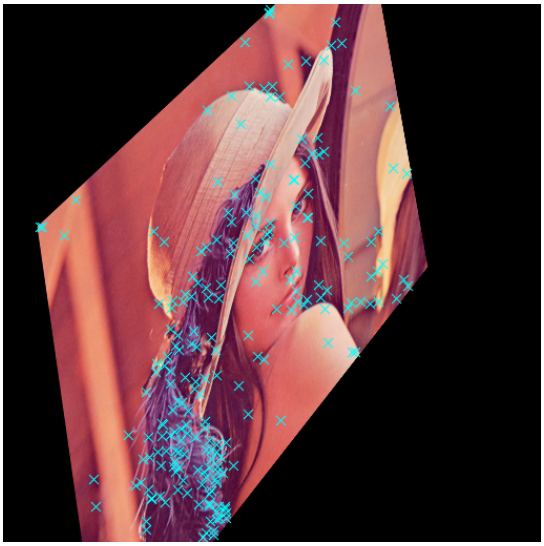
Least Median of Squares Least Median of Squares (LMS) [226] is another widely used robust model estimation technique. It operates very similarly to RANSAC. Instead of counting the number of inliers on the basis of whether the corresponding errors are within a predefined threshold, it finds the median value of these errors, and then tries to minimise this resultant median value. Compared to RANSAC, it fits outliers for robust regression as a minimisation problem with a nonlinear loss function, which results in it being necessary to employ a numerical optimisation algorithm to solve the regression problem [224]. It performs poorly when more than 50% of the features are outliers, but it has the advantage that there is no need to choose any threshold for inliers.

PROgressive SAmple Consensus PROgressive SAmple Consensus (PROSAC) [227] is an enhanced version of RANSAC. It speeds up the process of finding a statistically good set of inliers by adding randomly selected feature correspondences from the most confident match candidates. The confidence here is ordered by a similarity measure, and it is based on the assumption that candidates with high similarity scores are more likely to be inliers compared to those with low similarity scores. In practice, good hypotheses are often generated early in the process, which results in PROSAC achieving significant savings in the computational cost. However, although PROSAC often dramatically reduces the number of hypotheses required for testing, this depends on the data and also on the existence of a similarity measure to rank feature correspondences.

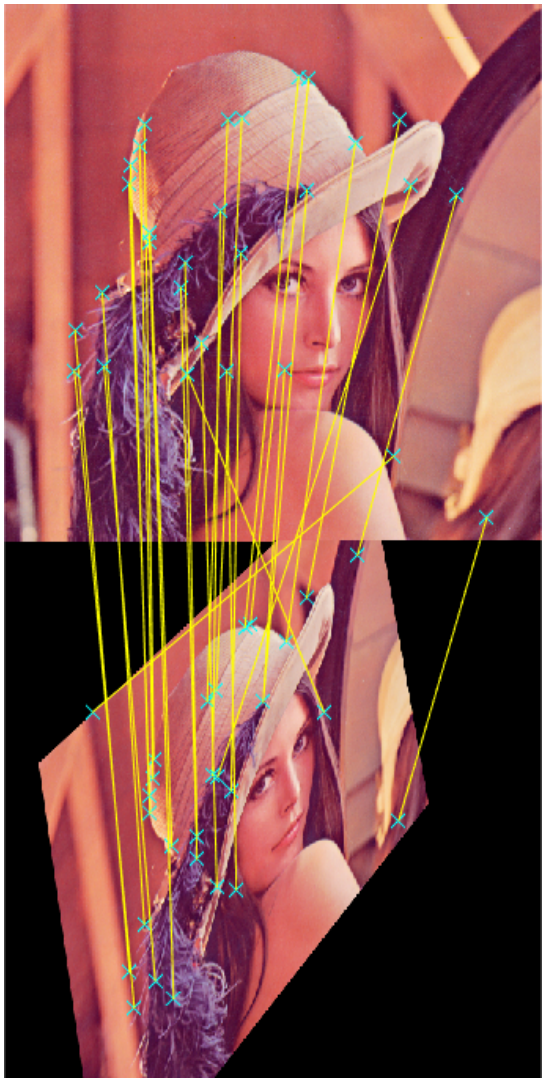
3.4.3 Estimating the Transformation Model

Once a set of matched feature correspondences has been established, the next step is to estimate the parameters of the transformation model that best registers the two images. In other words, the corresponding features from the reference and sensed images should be as close as possible after the sensed image is transformed by employing the estimated transformation model. The estimation of transformation models should take consideration of the assumed geometric distortion of the sensed image, the method of image acquisition, and the required accuracy of the final registration result. In general, the number of matched feature correspondences is often many more than the minimum number required for determining the transformation model, e.g. the projective deformation must be determined by at least 4 matched feature correspondences. This results in the estimation of transformation models becoming overdetermined problems, i.e. there are more equations than unknowns.

(a)



(b)



(c)

(d)

(e) registered result

(f)

Figure 3.13: An example of registering a projective distorted image by SIFT and RANSAC - the reference and sensed images are shown in Figure 3.2a and 3.2f, respectively: (a) and (b) the SIFT features detected in the reference and registered images (marked as cyan ×), respectively (c) and (d) the feature correspondences detected before and after employing RANSAC, respectively (f) the difference between the reference and registered images.

Least Squares

Least Squares is a classical method that approximates the solution of an overdetermined problem [228, 229]. The parameters of the transformation model are estimated so that the sum of squared errors is minimised at the matched feature correspondences. For the most general case, i.e. the projective transformation, its equations (3.31) can be rearranged as:

$$-(h_{11}x + h_{12}y + h_{13}) + (h_{31}x + h_{32}y + 1)x' = 0$$

and

$$-(h_{21}x + h_{22}y + h_{23}) + (h_{31}x + h_{32}y + 1)y' = 0 \tag{3.67}$$

Because the coefficients of the homography matrix $\mathbf{H}$ appear linearly, to solve for $\mathbf{H}$ the above equations are rearranged to get the following:

$$\mathbf{a}_{h_x}^T \mathbf{h} = 0$$

and

$$\mathbf{a}_{h_y}^T \mathbf{h} = 0 \tag{3.68}$$

where

$$\mathbf{h} = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, 1]^T \tag{3.69}$$

$$\mathbf{a}_{h_x} = [-x, -y, -1, 0, 0, 0, xx', yx', x']^T \tag{3.70}$$

$$\mathbf{a}_{h_y} = [0, 0, 0, -x, -y, -1, xy', yy', y']^T \tag{3.71}$$

For a given set of matched feature correspondences, the following linear system of equations can be formed:

$$\mathbf{A}_h \mathbf{h} = 0 \tag{3.72}$$

where

$$\mathbf{A}_h = \begin{pmatrix} \mathbf{a}_{h_x}^T \\ \mathbf{a}_{h_y}^T \\ \vdots \end{pmatrix} \tag{3.73}$$

A least squares estimation can be obtained, which is the eigenvector corresponding to the minimum eigenvalue of $\mathbf{A}_h^T \mathbf{A}_h$.

Weighted Least Squares The above formulation of least squares uses all matched feature correspondences for estimating the parameters of transformation model, which is valid for the entire image. In other words, it assumes all matched feature correspondences are matched with the same accuracy, which discards local geometric distortions by averaging them equally over the entire image. This is not desirable [18]. It may often result in poorly handled locally deformed images, i.e. unnecessarily warping some areas of the sensed image, where the matched feature correspondences are away from each other when aligned to the reference image. To avoid this problem, a special case of generalised least squares, called weighted least squares, is often employed [18, 135]. It is associated with an estimation of variance with each matched feature correspondence.

3.5 Area Based vs Feature Based

Since two alternative approaches exist to register images (i.e. the area based and the feature based), the question is: which is preferable for practical applications? Unfortunately, there is no clear answer for all situations.

When the images do not contain a lot of notable details, and the information provided by image intensity values are more distinctive compared to that provided by image local shapes/structures, the area based techniques are usually preferable [18]. For example, for registering typical medical images that are not very rich in details, and for matching sequential image frames in a video obtained from remote sensing, the area based approach can often work well. These techniques have the advantage of measuring and properly weighting the contribution of every pixel location in the image, so that all the information available in the image is used optimally [135]. They also have two principal limitations. One is that there must be either an identical or a statistically dependent similarity relationship between the intensity values of the reference and sensed images, so that correlation-like methods can be employed [18]. The other is that they usually only allow changes in shift and rotation to exist between the images

to be registered, which results in making them fail too often when registering partially overlapping images [135].

If the images contain enough distinctive objectives that can be easily detected (i.e. the information carried by image local structures is more significant than that carried by the image intensity values), the feature based methods are usually recommended [18]. For example, for the applications in computer vision, the images typically contain many details such as buildings, roads, rivers, and so on. As long as the distribution of detected features over the image is good, enough matched feature correspondences can usually be found [123]. However, the early feature based methods tend to work improperly in regions that are either too textured or not textured enough [135]. This often results in the obtained features being distributed unevenly over the image. This causes failures to match features that should in fact be aligned [135]. Fortunately, the modern feature based approaches usually have remarkably robust detection and matching schemes. In such cases, not only regions whose Harris measure (i.e. the change of intensity value due to shifts in a local window) is high (i.e. corner points) [203, 204], but also blob-like regions [192], and uniform areas [230] are all treated as features. Because these methods operate in a scale space and additionally employ a dominant orientation, they can handle complex distortions between images, e.g. changes in scale, orientation, as well as foreshortening (i.e. an object appears shorter than it actually is) due to homographies, and even allow the registration of images taken from widely separated views [192]. There are two crucial points for all feature based approaches. One is whether the respective features are stable and/or can be detected [18]. The other is whether the feature descriptors are constructed reasonably, so that there is a sufficient number of feature correspondences can be found to permit image alignment [135].

More recently, image registration methods have started to appear that employ both area based and feature based approaches simultaneously [231]. First, two images are aligned using feature based method; then, the sensed image is warped to the reference image frame; finally, a more accurate estimate can be recomputed by employing an area based method [135].

3.6 Image Resampling Techniques

In image registration, and many other image processing applications, images may be transformed (e.g. resized, rotated and distorted) to fractional pixel addresses/locations. To address the effect of the fractional pixel addresses, a number of interpolation algorithms have been proposed. An interpolation method estimates the values at unknown points by using the intensity values from known neighbouring pixels. Nevertheless, this is only an approximation. The resultant image always loses some quality when interpolation is performed, and resultant image can vary significantly depending on the interpolation algorithm. There are three most commonly used interpolation techniques,

i.e. nearest-neighbour, bilinear, and bicubic.

Nearest-Neighbour Interpolation Nearest-neighbour interpolation [232] is one of the simplest interpolation algorithms, where it only considers a single pixel, i.e. the nearest one to the interpolated point, to assigns the intensity value of this pixel to the interpolated point. Due to the simplicity of the process, nearest-neighbour interpolation outperforms others in terms of the processing time. However, this process also has the effect of simply making each pixel bigger, which can result in spatial distortion, and therefore makes it unreliable for measurement purposes.

Bilinear Interpolation

Bilinear interpolation [232], also known as bilinear filtering or bilinear texture mapping, considers the closest 2×2 neighbourhood pixel values surrounding the unknown pixel. It then takes a weighted average of these 4 pixels to obtain the final interpolated value. This results in much smoother looking images than the nearest neighbour method, and reducing the visual distortion caused by the fractional zoom calculation, which changes the size of an image while attempting to retain good image quality.

Suppose the function $f(\cdot)$ can represent the intensity value of a function, then given the four pixels $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$ and $Q_{22} = (x_2, y_2)$ surrounding the unknown fractional pixel $P(x, y)$, with the intensity values $f(Q_{11})$, $f(Q_{11})$, $f(Q_{11})$, $f(Q_{11})$ and $f(Q_{11})$, the linear interpolations in the x direction are carried out to get the values at R_1 and R_2, as:

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \tag{3.74}$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \tag{3.75}$$

then, linear interpolations in the y direction are:

$$\begin{aligned} f(P) &\approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \\ &\approx \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} f(Q_{11}) + \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} f(Q_{21}) \\ &\quad + \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} f(Q_{12}) + \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} f(Q_{22}) \end{aligned} \tag{3.76}$$

Bicubic Interpolation Bicubic interpolation [232] is more computationally demanding than nearest-neighbour or bilinear interpolation. It is similar to bilinear interpolation but the pixel intensity values are modelled by filling a cubic rather than a linear function, which requires a slight larger region to fit the function. It considers the closest 4×4 neighbour pixels of the interpolated point, and then takes a weighted average of these 16 pixels to get the final interpolated value, with closer pixels given a

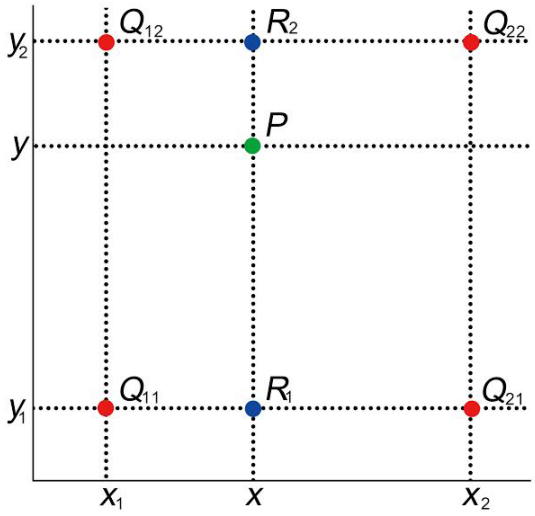


Figure 3.14: An example of interpolating intensity value by bilinear interpolation.

higher weighting in the calculation. It produces noticeably sharper images than bilinear and nearest-neighbour interpolations, with less blurring of edges and other distortion artifacts.

3.7 Image Registration in 3D Reconstruction

There exist many tools and techniques that attempt to obtain information about the geometry of a 3D scene from its 2D images. Because the image formation process is not generally invertible, the reconstruction problem needs to be solved with additional information to reduce the number of degrees of freedom, e.g. parallelism and coplanarity constraints that exploit prior knowledge about the scene. One possible approach to reconstruct a 3D point is by triangulation, if corresponding image points in two or more views are given. However there is an important prerequisite, i.e. the determination of a projection matrix, which expresses calibration and pose of camera. Fortunately, a technique termed Structure from Motion (SfM) can solve 3D points and projection matrices simultaneously and automatically, using an iterative bundle adjustment procedure based on corresponding points in multiple views with overlapping and offset [24]. It is most suitable for images that have a high degree of overlap and are derived from a wide range of positions, e.g. images captured by a moving sensor, so that the full 3D structure of the scene can be captured.

The key challenge before initialising the SfM process is registering images taken from different angles [24, 233], i.e. establishing the coordinate relationships between pairs of images, based on matching the features identified in each individual image that are potential candidates for feature correspondences. However, early techniques for image registration are only applicable when scenes are seen from similar viewpoints

(i.e. for narrow baseline matching), because they work by matching image patches located around the detected keypoints. Fortunately, methods for both narrow and wide baseline matching have been developed based on local image descriptors, and one popular approach is by matching SIFT features [23]. In the feature detection step, SIFT first automatically extracts keypoints over all locations and scales in each image, and then creates the corresponding feature descriptors in vector representations, based on local image gradients [192]. These computed descriptors are unique enough to be distinguished from each other, and they are largely insensitive to variations in the image scaling and rotation as well as partially invariant to changes in illumination condition and 3D viewpoint [192], so that the corresponding features can be matched confidently. In the feature matching step, the Approximate Nearest Neighbour (ANN) k-d tree [234] is usually used to match keypoints between each pair of images. To work more efficiently, a priority search algorithm for ANN only allows each query to test at most 200 bins in the tree. False matches are often identified by using a ratio test described in [192]. Further, RANSAC [223] is used to robustly estimate a fundamental matrix for the pair of images, while improving the robustness to noise and outliers [235]. To minimise errors for all the inliers to the estimated fundamental matrix, the estimated matrix can be refined by using the Levenberg-Marquardt Algorithm (LMA) [236]. LMA is a popular algorithm for constructing a mathematical function, so that a set of data samples can be best fitted to it. Finally, a set of geometrically matched feature correspondences are found between each pair of images, and are organised into tracks [24]. A track represents a set of matched keypoints that are connected across two or more images [24].

3.7.1 Structure from Motion

When a sufficient number of tracks are obtained, the Structure from Motion (SfM) technique can be employed to estimate camera poses and extract a sparse point cloud. In other words, it is used to recover a projection matrix and a 3D location for each image and track, respectively [24]. Projection matrices can be estimated through recovering a set of camera parameters, e.g. camera rotation and translation [237, 238], while 3D locations can be estimated by the technique called triangulation [239]. The recovered values are only initial estimates, and usually they need to be refined by employing an iterative non-linear optimisation technique, called bundle adjustment [25] (it will be described below). Bundle adjustment aims at optimising the 3D structure and viewing parameters, so that an optimal reconstruction is obtained under certain assumptions. After utilising bundle adjustment, the optimal values can be obtained by minimising the reprojection error, i.e. the sum of differences between the projections of each track and the corresponding keypoints in each image [24]. But the bundle adjustment technique only guarantees to find local minima, then it may fail by converging to bad

local minima (particularly, for SfM problems with large-scale data). To avoid this, a suitable initialisation of the parameter values must be provided [24].

Usually, there are two different approaches to SfM: incremental SfM, and global SfM. The incremental SfM approach is the most popular, where successive views are incorporated one at a time to solve camera poses and 3D locations. Typically, good initial estimates of parameter values can be obtained by utilising the first two views. There are also two important limitations: One is that a large number of feature correspondences must be defined in each view, which requires considerable overlaps. For long sequences of with many views, this requirement is hard to meet. The other is that there exists various kinds of degenerate structure and motion configuration which are almost impossible to recover [240], or multiple regions have the same structure so that they cannot be distinguished. Degenerate configurations imply the situation where there is an insufficient number of data samples available to determine a unique solution [241]. In practice, it may be hard to avoid degeneracy when views are obtained without careful planning. Unlike the incremental method, the global SfM approach works by computing parameter values based on utilising all available views simultaneously. It has advantages over the incremental method by avoiding gross errors associated with the sequence closure, i.e. adding images one by one. This is achieved by distributing reconstruction errors meaningfully across all measurements. It shares the same limitation as the incremental approach, i.e. there exist degenerate structures and motion configurations that may fail to recover the structure. In addition, the global SfM is not able to deal with missing data, i.e. every 3D point must be available in every view [242].

Recovering Projection Matrices

The camera parameters (the camera projection matrices) can be recovered by decomposing a fundamental matrix, which depends on the relative position and orientation of a pair of images. The estimation of a fundamental matrix can be achieved when there are sufficient feature correspondences available, and the resultant estimation is accurate regardless of noise present in images [243] and robust to the existence of outliers [244, 245].

Estimating The Fundamental Matrix The fundamental matrix $\mathbf{F}_e$ [131] is a 3×3 matrix that relates a pair of views, but it only has a rank of 2 (i.e. it is not invertible), because it has a null space that is not just the zero vector [246]. For a feature point (x, y) in the image I and its corresponding point (x', y') in the image I', they should satisfy an epipolar constraint, i.e. (x', y') must lie on the particular epipolar line generated by (x, y) and the epipolar. The epipolar is defined as the intersection of the line joining the baseline with the image plane [26]. The epipolar constraint can be formulated algebraically with the use of homogenous coordinates and the fundamental matrix $\mathbf{F}_e$,

as [131]:

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \tag{3.77}$$

When n pairs of feature correspondences exist, the above equation can be rearranged in the representation of linear equations in the 9 unknown elements of the fundamental matrix $\mathbf{F}_e$, as [131]:

$$\begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \mathbf{0} \tag{3.78}$$

or in matrix format as:

$$\mathbf{A}_f\mathbf{f} = \mathbf{0} \tag{3.79}$$

where $\mathbf{A}_f$ is an $n \times 9$ matrix representing the n pairs of corresponding feature points, and $\mathbf{f}$ is a 9-d vector storing all the 9 elements of the fundamental matrix $\mathbf{F}_e$. A least square solution can be found, as the eigenvector corresponding to the minimum eigenvalue of $\mathbf{A}_f^T\mathbf{A}_f$. This solution to $\mathbf{f}$ is not unique [247]. It is defined up to an arbitrary scale. Because the fundamental matrix $\mathbf{F}_e$ has a rank of 2, it has only 7 degrees of freedom. It is important to pre-process the corresponding feature points, otherwise the computation can be poorly conditioned [243]. The pre-processing process can be obtained by normalising these points, so that the condition number of $\mathbf{A}_f^T\mathbf{A}_f$ (i.e. the ratio of the largest to smallest singular value in the SVD of $\mathbf{A}_f^T\mathbf{A}_f$) is improved before estimating the elements of the fundamental matrix $\mathbf{F}_e$ [243].

For a particular combination of camera and lens, the camera intrinsic and extrinsic parameters can be determined by taking photographs of a controlled scene. Then it is reasonable to assume that the camera calibration matrices $\mathbf{K}_{\text{cam}}$ and $\mathbf{K}'_{\text{cam}}$ are known for the image I and I', respectively. The above recovered fundamental matrix $\mathbf{F}_e$ can be transformed into an essential matrix through the following equation:

$$\mathbf{E} \sim {\mathbf{K}'_{\text{cam}}}^T\mathbf{F}_e\mathbf{K}_{\text{cam}} \tag{3.80}$$

where $\mathbf{E}$ is a 3×3 matrix, called the essential matrix, which describes the relationships between corresponding feature points in a pair of images. The estimates of the translation and rotation between the two views can be obtained, by further decomposing the resultant essential matrix $\mathbf{E}$ into a skew-symmetric matrix and an orthonormal matrix, respectively, as:

$$\mathbf{E} \sim [\mathbf{T}]_\times\mathbf{R} \tag{3.81}$$

where $[\mathbf{T}]_\times$ is a 3×3 matrix, called the cross product matrix. For a translation vector $\mathbf{T} = [t_x, t_y, t_z]^T$, the corresponding cross product matrix $[\mathbf{T}]_\times$ is defined by:

$$[\mathbf{T}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \tag{3.82}$$

and $\mathbf{R}$ is a 3×3 matrix, corresponding to a rotation.

The above decomposition can be achieved based on computing the Singular Value Decomposition (SVD) of the essential matrix $\mathbf{E}$ [248]:

$$\mathbf{E} = \mathbf{U}_e \mathbf{\Lambda} \mathbf{V}_e^T \tag{3.83}$$

where $\mathbf{U}_e$ and $\mathbf{V}_e$ are 3×3 orthogonal matrices, and $\mathbf{\Lambda}$ is a 3×3 diagonal matrix. The translation $\mathbf{T}$ and rotation $\mathbf{R}$ can then be obtained by:

$$[\mathbf{T}]_\times = \mathbf{U}_e \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{U}_e^T \tag{3.84}$$

$$\mathbf{R} = \mathbf{U}_e \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{V}_e^T \tag{3.85}$$

Finally, the projection matrices $\mathbf{P}$ and $\mathbf{P}'$ can be estimated directly from the recovered translation $\mathbf{T}$ and rotation $\mathbf{R}$ above, as:

$$\mathbf{P} = \mathbf{K}_{\text{cam}}[\mathbf{I} \mid \mathbf{0}] \tag{3.86}$$

$$\mathbf{P}' = \mathbf{K}'_{\text{cam}}[\mathbf{R} \mid \mathbf{T}] \tag{3.87}$$

where $\mathbf{I}$ is a 3×3 identity matrix. However, due to the arbitrary choice of signs for the translation $\mathbf{T}$ and rotation $\mathbf{R}$, there are four possible solutions. But because the reconstructed 3D points must lie in front of the cameras, the correct one can be easily disambiguated. However, the recovered projection matrices are still ambiguous in scale, more specifically, the camera translation corresponds to an unknown scale.

Triangulation

Triangulation [239] refers to the process of computing the location of a 3D point, given its projections in two or more views, as well as the camera projection matrices describing the projection process from 3D to 2D for all the cameras involved. In an image, each feature point and the corresponding camera location can be used to generate a line in 3D space that this point lies on. Additionally, the corresponding feature points in two or more images must be the projections of the same 3D point. The coordinate of this 3D point can be recovered by computing the intersection point of the set of lines

generated based on the corresponding feature points. However, due to the presence of measurement noise, the above ideal situation will not generally hold, i.e. the set of lines usually will not intersect. Thus, the location of the 3D point has to be estimated using an alternative method, in such a way as to minimise an appropriate error metric. The most commonly used one is the sum of squared errors between the measured and predicted feature point locations of the 3D point in all the images where it is visible, as [239]:

$$\arg\min_{\mathbf{X}_{3\mathrm{D}}} \sum_i \|\mathbf{Y}_{2\mathrm{D}_i} - \mathbf{P}_i \mathbf{X}_{3\mathrm{D}}\|_2^2 \tag{3.88}$$

where $\mathbf{Y}_{2\mathrm{D}_i} = [y_{1_i}, y_{2_i}, 1]^T$ and $\mathbf{P}_i\mathbf{X}_{3\mathrm{D}}$ are the measured and predicted feature point positions of the 3D point $\mathbf{X}_{3\mathrm{D}} = [x_1, x_2, x_3, 1]^T$ in the image i, and $\mathbf{P}_i$ is the 3×4 camera projection matrix corresponding to the view i. Under the assumption that the measurement noise associated with the image coordinate is Gaussian distributed, a solution for the 3D point $\mathbf{X}_{3\mathrm{D}}$ can be obtained using maximum-likelihood estimation [239].

Another popular approach works by exploiting the property of parallelism between the homogenous 3-D vectors $\mathbf{Y}_{2\mathrm{D}_i}$ and $\mathbf{P}_i\mathbf{X}_{3\mathrm{D}}$, which can be used to form the following equation [239]:

$$[\mathbf{Y}_{2\mathrm{D}_i}]_\times \mathbf{P}_i \mathbf{X}_{3\mathrm{D}} = \mathbf{0} \tag{3.89}$$

Since there is a number of images where the reconstructed 3D point $\mathbf{X}_{3\mathrm{D}}$ is visible, the above equation can be arranged into matrix form, as:

$$\mathbf{A}_p \mathbf{X}_{3\mathrm{D}} = \mathbf{0} \tag{3.90}$$

where $\mathbf{A}_p$ is a $3n \times 4$ matrix that is formed by:

$$\mathbf{A}_p = \begin{bmatrix} [\mathbf{Y}_{2\mathrm{D}_1}]_\times \mathbf{P}_1 \\ [\mathbf{Y}_{2\mathrm{D}_2}]_\times \mathbf{P}_2 \\ \vdots \\ [\mathbf{Y}_{2\mathrm{D}_n}]_\times \mathbf{P}_n \end{bmatrix} \tag{3.91}$$

where n represents the total number of views where the 3D point $\mathbf{X}_{3\mathrm{D}}$ is visible. Further (3.90) above can be transformed into an optimisation problem, whose solution can be approached by the least squares method, as:

$$\arg\min_{\mathbf{X}_{3\mathrm{D}}} \|\mathbf{A}_p \mathbf{X}_{3\mathrm{D}}\|_2 = 0 \tag{3.92}$$

Finally, the solution to the 3D point $\mathbf{X}_{3\mathrm{D}}$ can be obtained as the eigenvector of $\mathbf{A}_p^T\mathbf{A}_p$ corresponding to the smallest eigenvalue.

Bundle Adjustment

After the parameters of camera motion and scene structure have been estimated, they need to be refined iteratively by employing a technique called bundle adjustment [25].

The bundle adjustment technique attempts to achieve an optimal estimate of a set of parameters, when the data points are noisy. It works by minimising an appropriate cost function. In particular, for 3D reconstruction, it is based on a weighted sum of squared reprojection errors. For practical applications, the Gauss-Newton algorithm is usually employed for fast convergence [25].

Since neither the parameters of projection matrices nor the 3D point locations can be observed directly, the measured pixel locations of the projected 3D points are used. Let the set of predicted pixel locations be $\bar{\mathbf{x}}_i(\theta)$ and the corresponding set of observed ones be $\hat{\mathbf{x}}_i$, where θ represents a set of parameters that need to be optimised. Then the prediction error $\Delta\bar{\mathbf{x}}_i(\theta)$ can be given by:

$$\Delta\bar{\mathbf{x}}_i(\theta) = \hat{\mathbf{x}}_i - \bar{\mathbf{x}}_i(\theta) \tag{3.93}$$

Under the assumption that the measurement noise is Gaussian distributed, the appropriate cost function for bundle adjustment to minimise is the sum of the squared prediction errors, which is based on the maximum-likelihood estimation, as:

$$f_p(\theta) = \frac{1}{2}\sum_i \Delta\bar{\mathbf{x}}_i(\theta)^T \mathbf{W}_i^p \Delta\bar{\mathbf{x}}_i(\theta) \tag{3.94}$$

where $\mathbf{W}_i^p$ is a positive definite weight matrix that is chosen to approximate the inverse covariance of the measurement noise associated with $\hat{\mathbf{x}}_i$.

The minimisation process of the above equation (3.94) is iterative. At each iteration, a parameter displacement $\theta \to \theta + \delta\theta$ is seek to minimise $f_p(\theta)$. It is possible to solve an approximate value for $\delta\theta$, by fitting a quadratic Taylor series to the cost function as [236]:

$$f_p(\theta + \delta\theta) \approx f_p(\theta) + \mathbf{g}_v^T \delta\theta + \frac{1}{2}\delta\theta^T \mathbf{H}_m \Delta\delta\theta \tag{3.95}$$

where $\mathbf{g}_v$ and $\mathbf{H}_m$ are the gradient vector and Hessian matrix, respectively. Under the assumption that the Hessian matrix $\mathbf{H}_m$ is positive definite, the above equation has a unique global minimum. Then the stationary point can be obtained by setting $\frac{df_p(\theta+\delta\theta)}{d\theta} \approx \mathbf{H}_m\delta\theta + \mathbf{g}_v$ to zero, which results in the step in Newton's method as:

$$\delta\theta = -\mathbf{H}_m^{-1}\mathbf{g}_v \tag{3.96}$$

The gradient vector $\mathbf{g}_v$ and Hessian matrix $\mathbf{H}_m$ can be obtained in terms of the Jacobian matrix $\mathbf{J}_m = \frac{d\bar{\mathbf{x}}(\theta)}{d\theta}$, by differentiating the equation (3.94) as:

$$\mathbf{g}_v = \frac{df_p(\theta)}{d\theta} = \mathbf{J}_m^T \mathbf{W}^p \Delta\bar{\mathbf{x}} \tag{3.97}$$

$$\mathbf{H}_m = \frac{d^2 f_p(\theta)}{d\theta^2} = \mathbf{J}_m^T \mathbf{W}^p \mathbf{J}_m + \sum_i (\Delta\bar{\mathbf{x}}^T \mathbf{W}^p)_i \frac{d^2(\Delta\bar{\mathbf{x}})_i}{d\theta^2} \tag{3.98}$$

where $\mathbf{W}^p = \mathrm{diag}(\mathbf{W}_1^p, \mathbf{W}_2^p, \ldots, \mathbf{W}_N^p)$, and the notation $(\ldots)_i$ represents the ith element of a vector. In practice, the term $\frac{d^2(\Delta\bar{\mathbf{x}})_i}{d\theta^2}$ in the Hessian matrix $\mathbf{H}_m$ is likely to be

small when comparing to the term $\mathbf{J}_m^T\mathbf{W}\mathbf{J}_m$ [25]. Then the least squares approximation to the Hessian matrix $\mathbf{H}_m$ can be obtained, by dropping this term as:

$$\mathbf{H}_m \approx \mathbf{J}_m^T\mathbf{W}^p\mathbf{J}_m \tag{3.99}$$

Finally, a Gauss-Newton approximation for $\delta\theta$ can be achieved as:

$$(\mathbf{J}_m^T\mathbf{W}^p\mathbf{J}_m)\delta\theta = -\mathbf{J}_m^T\mathbf{W}^p\Delta\bar{\mathbf{x}} \tag{3.100}$$

The Gauss-Newton approximation can get a very accurate result, when the least square cost function is free of outliers and evaluated near the real cost minimum [25].

The last challenge left is the determination of the time to stop the iteration, which progressively obtains a better estimation of the parameters. Since the solution obtained by optimising the minimisation problem of a cost function is only a statistical approach to the true solution, it is usually unnecessary and wasteful to iterate until a limitation is reached (e.g. converging to a machine accuracy). A simple but effective solution is descried in [249]. The iteration is stopped if there are two successive situations, where the amount that the cost function (i.e. $f_p(\theta)$) decreases is less than a small fractional number (e.g. 10^{-3}).

After the above stages, a sparse point cloud is produced. To enhance the density of the point cloud, a number of techniques have been developed. For example, the combination of the Clustering Views for Multi-view Stereo (CMVS) [250, 251] and Patch-based Multi-view Stereo (PMVS) algorithms [250], which works based on the camera positions derived from SfM. First, the CMVS technique is employed to determine which images are overlapping, and further decompose these overlapping images into clusters or subsets whose size is manageable. Then the PMVS technique is used to reconstruct the 3D information from each of these resultant clusters independently. Finally, a significant increase in the density of the point cloud can be achieved by employing the above process, typically being around two orders of magnitude [23].

The estimation results obtained from the SfM procedure is in relative coordinates, i.e. they are ambiguous to the scale. However, sometimes the resultant model need to be displayed on an overhead map, then its coordinates must be transformed to absolute coordinates, which can be achieved by aligning the model with a geo-referenced image or map, e.g. a satellite image [24]. The algebraic relationship between relative and absolute coordinates can be represented by a similarity transform. The determination of the correct transformation can be achieved by interactively rotating, scaling, and translating the resultant model, until an agreement is obtained between the model and the provided image or map [252].

Chapter 4

Binary Data Embedding Framework

To be able to generalise properties of the data from very limited sets of examples. A novel manifold embedding method is proposed to generate binary decision processes, where the unique characteristics of each individual class are determined and a low dimensional embedding is constructed. The proposed method outperforms other methods even when only small numbers of examples are available for training. This chapter is mainly based on the published work [31, 32].

4.1 Motivation

There has been a rapid expansion in the number of video surveillance systems in everyday environments. The introduction of high quality digital cameras in public places provides image data for situational assessment and crowd monitoring, and specific security functions, such as the identification of individuals. Machine learning and other techniques have been developed to assist human operators and to reduce the workload associated with monitoring large numbers of cameras. For example, facial recognition algorithms are widely used in automated security systems (e.g. airport security). A computer processes a large number of digital images or video sequences to identify individuals who may pose security concerns or to authenticate people who are authorised to access a certain area [253]. A direct way to approach the face recognition problem is visual face recognition, i.e. 2D face recognition. Although this approach causes information loss, due to the projection from 3D to 2D, it can still achieve high performance, when face images are taken under certain controlled conditions [253]. To make the system more tolerant to environmental variations, such as changes in background, image resolution, pose, illumination, expression and occlusion, an alternative approach is full 3D face recognition, where the face is represented as a full 3D object. Because this approach includes the complete geometry of the face, it should be robust with respect to variations in pose and illumination [254, 255]. However, existing 3D

sensor technologies are not completely independent of lighting conditions, and can be affected by strong light sources [254]. A promising approach uses Infrared (IR) face recognition [256], which uses thermal differences arising from the tissues under the skin as well as external visual features, and it could work in total darkness [257]. It does however require special camera equipment.

Nevertheless, all three mentioned face recognition approaches can be represented as a generic classification problem [253]: to determine to which of a set of identities (i.e. classes) one or more new face images (i.e. data samples) should correspond, on the basis of an existing set of face images whose corresponding identity (i.e. label) information is known. The input data frequently possess high dimensionality [258, 259]. This is due to the fact that increasing data dimensions (e.g. using high resolution face images) can provide more information about objective phenomenon [260], and thus can improve the performance of a variety of learning tasks [261, 262]. However, the number of available data samples is usually limited. Therefore, high dimensional data often have many more degrees of freedom than can be defined and/or determined for a perceptually meaningful structure for the purpose of identification or verification [12, 13, 93]. Thus, directly processing the data in its original high dimensional feature space would lead many existing classification algorithms to increase computational complexity [263], lack efficiency, or even fail [264].

The intrinsic degrees of freedom often imply that there is a low dimensional structure embedded in the original feature space of the high dimensional data [264]. A direct way to achieve low dimensional representation is by feature selection. It aims to remove redundant and/or irrelevant information by selecting a subset of features from the original data so that the redundant and/or irrelevant features do not degrade the classification performance. It is easy to interpret the results, but searching all possible feature subsets creates a relatively high computational cost. Additionally, it may cause information loss due to the selection process. Feature extraction is an alternative approach, where it projects the data from its original high dimensional feature space to a lower one. The resultant features are either a linear or nonlinear transformation of the original features. Unlike to feature selection, it rotates the coordinate system of the original high dimensional data, and then selects a number of features that are potentially important [260]. Embedding learning is a group of feature extraction techniques which aim to learn a compact and meaningful representation of a set of data samples in a high dimensional feature space.

A range of algorithms for embedding learning have been proposed [15, 16], and they have been reviewed in detail in chapter 2. Most algorithms differ from one another in how, during the learning process, they preserve important properties and characteristics that are present in the original set of data. They use different approaches to preserve similar structural information, based on formulating different objective functions, along

with different constraints for trace optimisations. The different objective functions can be optimised using an eigen solver [265–267], but their underlying goals are similar. Each attempts to find one embedded feature space that groups the data samples of intraclass together, whilst maintaining a suitable separation between those of interclass. In real world applications, when a sufficient number of data samples is available, most of the existing algorithms can achieve good performance. However, it does not often hold (usually, there is only a small number of data samples are available to characterise the underlying structure of a given dataset) and then it is very difficult for these algorithms to achieve good performance.

To try to overcome this problem, a new approach is proposed to aim at generating a feature space formed by distinct feature(s) of each class, which could potentially enhance the grouping of the data samples in this class while separating them from those in the remaining classes. Compared to the existing techniques that try to find distinguishing features, each of which attempts to separate all classes, this approach focuses on finding the distinct characteristics of one class at a time. The proposed binary data embedding framework is motivated by the idea of One-Vs-All (OVA) [27, 28], which is a common strategy that decomposes a multiclass classification problem into several binary ones [268, 269]. This is based on the fact that binary decomposition shows better performance compared to other multiclass approaches [270]. The issue of an imbalanced number of data samples is usually associated with binary decomposition, i.e. the number of data samples from a target class is relatively small, compared to that of the remaining data samples. It can cause some undesirable effects in the derived performance [271–273]. To address the issue, instead of selecting data samples from the remaining ones to achieve a balance, a weighted pairwise constraint indicator is proposed to balance the contributions of the data samples from a target class from those from the remaining classes.

4.2 Mathematical Formulation

The input set of data samples often possesses very high dimensionality, which may contain redundant and/or irrelevant information. This results in increasing computational complexity. To avoid this, a new set of data samples can be obtained through data pre-processing. It aims to reduce the original dimensionality of the input set of data samples, whilst keeping the essential characteristics and properties (e.g. the data variance) of the original set. The way to apply a data pre-processing technique to both training and test set $\mathbf{X}$ and $\widetilde{\mathbf{X}}$ is listed in appendix B. Subsequently, a corresponding pairwise feature based weight is computed, to represent the relation features (i.e. being similar or dissimilar) between data samples of pre-processed set of data samples. The way to compute a corresponding pairwise feature based weight matrix $\mathbf{F} = [f_{ij}]$ for the training set $\mathbf{X}$ can be found in appendix A.

The underlying concept of the proposed method is the idea of formulating an embedded feature space that separates the data samples into either the target class, or belonging to the set of the remaining classes. The proposed method formulates the final objective embedded feature space, with the use of the pre-processed set of data samples and the pairwise feature based weight representing the relation features within the pre-processed set of data samples.

The results generated by most of the existing embedding techniques for classification can be viewed as modifications of the relative positions between data samples in the embedded feature space. One common way to achieve this is to optimise the following minimisation problem [14]:

$$\arg\min_{\mathbf{z}_i \in R^b} \sum\nolimits_{i,j=1}^{n} w_{ij} \left\| \mathbf{z}_i - \mathbf{z}_j \right\|_2^2 \tag{4.1}$$

where the weight w_{ij} controls the relative position between embedded data samples $\mathbf{z}_i$ and $\mathbf{z}_j$ [60,62]. w_{ij} can be divided into two meaningful terms: its sign indicates different ways to modify the relative position between $\mathbf{z}_i$ and $\mathbf{z}_j$, i.e. moving far apart (a negative value), staying close together (a positive value) and no action (a value of 0); while its absolute value indicates how important the corresponding position modification is. Then, to characterise these two meaningful terms, w_{ij} can be expressed as:

$$w_{ij} = \mathrm{sgn}(w_{ij}) |w_{ij}| \tag{4.2}$$

where $\mathrm{sgn}(\cdot)$ is the sign function, and it is defined as:

$$\mathrm{sgn}(x) = \begin{cases} +1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{4.3}$$

while $|\cdot|$ indicates the absolute value, and it is defined as:

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases} \tag{4.4}$$

The above three different ways of position modification can be characterised with a pairwise constraint [274, 275]. Because they utilise feature proximities to generate results, for the unsupervised embedding techniques, there is no position modification that moves data samples far apart. Additionally, to incorporate local neighbour graphs [54], a parameter k is introduced to indicate the k-NN (Nearest Neighbour) for each data sample. Then, one possible indicator for unsupervised pairwise constraint δ_{ij}^u can be defined as:

$$\delta_{ij}^u = \begin{cases} +1 & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are undirected } k\text{-NN} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

For the methods that do not take into account the local neighbour information, such as PCA [42], δ_{ij}^u is still applicable by setting k to be $n-1$ (i.e. treat all the available

data samples to be its nearest neighbours). While for the supervised techniques that additionally utilise the label information, to achieve better grouping and separation of respective intraclass and interclass data samples, they aim to keep friend data samples (i.e. ones from the same class) close together while enemy ones (i.e. ones from different classes) far apart (the terms friend and enemy to indicate respective intra- and inter-class were used in some previous work, e.g. [14]). Similarly, the local neighbour information can be characterised by introducing two parameters k_F and k_E, to indicate k_F-NN and k_E-NN among friend and enemy data samples, respectively. Then, one possible indicator for supervised pairwise constraint δ_{ij}^s can be defined as [14]:

$$\delta_{ij}^s = \begin{cases} +1 & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are from the same class} \\ & \text{and undirected } k_F\text{-NN} \\ -1 & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are from different classes} \\ & \text{and undirected } k_E\text{-NN} \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

Similarly, for the methods that do not take into account the local neighbour graphs, such as MMC [51], δ_{ij}^s is still applicable by setting k_F and k_E to be certain values, so that all friend and enemy data samples are included.

Some embedding techniques permit all data samples to make the same contribution to the calculations, by setting $|w_{ij}|$ to be a constant value, such as PCA [42]. More techniques operate on contributions from selected pairwise data samples (e.g. pairwise undirected nearest neighbours), and the contributions are distinguished to be strong or weak according to a similarity measure based on the corresponding features. For the supervised methods that incorporate local neighbour information, it is reasonable to employ a feature based weight f_{ij} to replace $|w_{ij}|$, and the corresponding optimisation problem in (4.1) can be reformulated as:

$$\arg\min_{\mathbf{z}_i \in R^b} \sum\nolimits_{i,j=1}^{n} \delta_{ij}^s f_{ij} \left\| \mathbf{z}_i - \mathbf{z}_j \right\|_2^2 \tag{4.7}$$

Adaptive Measures of Intra- and Inter-class Information

The Fisher criterion [50] is a standard method that characterises label information, where it simultaneously minimises and maximises the measure of the within-class scatter $\mathbf{S}_w$ and that of the between-class scatter $\mathbf{S}_b$, respectively. The simultaneous maximisation and minimisation can be achieved by a trace optimisation [14, 265–267]. According to the work in [57], both trace$[\mathbf{S}_w]$ and trace$[\mathbf{S}_b]$ can be reformulated to be the forms that are similar to the objective function in (4.1), as:

$$\text{trace}[\mathbf{S}_w] = \sum\nolimits_{i,j=1}^{n} w_{ij}^w \left\| \mathbf{z}_i - \mathbf{z}_j \right\|_2^2 \tag{4.8}$$

$$\text{trace}[\mathbf{S}_b] = \sum\nolimits_{i,j=1}^{n} w_{ij}^b \left\| \mathbf{z}_i - \mathbf{z}_j \right\|_2^2 \tag{4.9}$$

where the weights w_{ij}^w and w_{ij}^b are defined as:

$$w_{ij}^w = \begin{cases} 1/n_t & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are from} \\ & \text{the same } t\text{th class} \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

$$w_{ij}^b = \begin{cases} 1/n - 1/n_t & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are from} \\ & \text{the same } t\text{th class} \\ 1/n & \text{otherwise} \end{cases} \tag{4.11}$$

While in [14], two measurable concepts friend closeness C_f and enemy dispersion D_e are introduced to provide an alternative way to describe the information of intraclass and interclass. One possible way to define C_f and D_e is as follows:

$$C_f = \sum\nolimits_{i,j=1}^{n} \delta_{ij}^s (\delta_{ij}^s + 1) f_{ij} \left\| \mathbf{z}_i - \mathbf{z}_j \right\|_2^2 \tag{4.12}$$

$$D_e = \sum\nolimits_{i,j=1}^{n} \delta_{ij}^s (\delta_{ij}^s - 1) f_{ij} \left\| \mathbf{z}_i - \mathbf{z}_j \right\|_2^2 \tag{4.13}$$

They can be viewed as divided the objective function in (4.7) into two separated parts, one for minimisation while the other for maximisation. Although they share similar forms as trace[$\mathbf{S}_w$] and trace[$\mathbf{S}_b$], they have the advantage of two factors. One is that they only accept the contributions from selective pairwise data samples (based on δ_{ij}^s), the other is that they weight these combinations to define the strong and weak data samples of friend and enemy (based on f_{ij}).

For the proposed method, an attempt is made to find an embedded feature space for a given dataset by treating each class individually, and then trying to find distinct feature(s) to separate a target class from the remaining classes. However, for binary classification, problems may arise when there is an unbalanced number of data samples. In such case, the number of class data samples should be reweighted, to avoid data samples from the target class being swamped by ones from the remaining ones. One possible solution to this problem is down selecting data samples from the class which has a larger number of data samples. However, it may be difficult to decide which data samples should be selected and, in addition, only using the selected data samples may fail to generalise the characteristics and properties of the class which the data samples are selected from. As a result, an alternative approach is proposed, which uses a weighted pairwise constraint indicator $\mathbf{\Gamma} = [\gamma_{ij}]$ that is similar to the indicator for supervised pairwise constraint $\mathbf{\Delta}^s = [\delta_{ij}^s]$ defined in (4.6), but additionally balance the contributions from the target class and the remaining classes.

As with most of the supervised embedding techniques for classification, the proposed method considers both local neighbour graphs and class structures of a given dataset to be of equal importance. For a target class, since the number of data samples is relatively small, all data samples in the target class should make contributions to the measure of the corresponding intraclass information. While for the remaining classes, a parameter k is employed to obtain selective contributions to the measure of the

intraclass information. To preserve some original grouping information within each of the remaining classes, the search of undirected k-NN is restricted to the data samples from one remaining class. For the measure of the interclass information, to incorporate local neighbour information, an undirected k-NN search is carried out between the data samples from the target class and those from the remaining classes. For the tth class ($t = 1, \ldots, c$), one possible way to define the corresponding weighted pairwise constraint indicator $\mathbf{\Gamma}^t = [\gamma_{ij}^t]$ is as follows:

$$\gamma_{ij}^t = \begin{cases} +\frac{k}{n_t} & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are from} \\ & \text{the same } t\text{th class} \\ -1 & \text{if either } \mathbf{x}_i \text{ or } \mathbf{x}_j \text{ is from} \\ & \text{the } t\text{th class while the other is from} \\ & \text{one of the remaining classes,} \\ & \text{and undirected } k\text{-NN} \\ +\frac{n_t-1}{n-n_t} & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are from} \\ & \text{one remaining class} \\ & \text{and undirected } k\text{-NN} \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

The weighted pairwise constraint indicator $\mathbf{\Gamma}^t = [\gamma_{ij}^t]$ above is designed based on the idea of assigning different weights to the pairs of data samples of the target class and to those of the remaining classes. The different weights would result in that the contributions from the target class and the remaining classes are balanced (i.e. equal in quantities). A simple way to achieve this balance is by setting the assigned weights to be in inverse proportion to the number of pairs of data samples. Here, the number of pairs of data samples in the target class that make contributions is $n_t \times (n_t - 1)$, while that in the remaining classes that make contributions is approximately $(n - n_t) \times k$. As a result, the weight of $+k/n_t$ is assigned to the pairs of data samples in the target class, while the weight of $+(n_t - 1)/(n - n_t)$ is assigned to those in the remaining classes, as formulated in (4.14).

In order to distinguish the stronger and weaker contribution that each data sample makes, the pairwise feature based weight matrix $\mathbf{F} = [f_{ij}^r]$ is employed. As a result, similar data samples from the same class correspond to a large weight f_{ij}. They are thus forced to stay even closer in the embedded feature space, which improves the within-class compactness. For the data samples from different classes which correspond to a large weight f_{ij}, it is very likely that they are both boundary points, thus they can be forced to move apart further, which improves the overall between-class separation. To separate the tth class from the remaining classes, two adaptive measures of the corresponding intraclass and interclass information are defined, as:

$$C_f^t = \sum\nolimits_{i,j=1}^{n} \text{sign}(\gamma_{ij}^t)[\text{sign}(\gamma_{ij}^t) + 1]\gamma_{ij}^t f_{ij} \left\|\mathbf{z}_i - \mathbf{z}_j\right\|_2^2 \tag{4.15}$$

$$D_e^t = \sum\nolimits_{i,j=1}^{n} \text{sign}(\gamma_{ij}^t)[\text{sign}(\gamma_{ij}^t) - 1]\gamma_{ij}^t f_{ij} \left\|\mathbf{z}_i - \mathbf{z}_j\right\|_2^2 \tag{4.16}$$

Optimisation of Adaptive Measures

Similar to the Fisher criterion [50], the two measures of C_f^t (i.e. the intraclass information) and D_e^t (i.e. the interclass information) are optimised by simultaneously minimising C_f^t while maximising D_e^t. The generated optimal embeddings should improve the separation between classes and the combination within each class, and additionally it should reduce dimensionality so that the computational complexity is reduced. Here the two trace optimisation templates suggested in [14] are employed which introduces a regularisation parameter to prevent overfitting, as:

$$\underset{\mathbf{V}\mathbf{V}^T=\mathbf{I}_{b\times b}}{\arg\max}\ \text{trace}\left[\mathbf{V}\mathbf{X}^T(\mathbf{A}_t-\lambda\mathbf{B}_t)\mathbf{X}\mathbf{V}^T\right] \tag{4.17}$$

or

$$\underset{\mathbf{V}(\mathbf{X}^T\mathbf{B}_t\mathbf{X}+\lambda\mathbf{I}_{d\times d})\mathbf{V}^T=\mathbf{I}_{b\times b}}{\arg\max}\ \text{trace}\left[\mathbf{V}\mathbf{X}^T\mathbf{A}_t\mathbf{X}\mathbf{V}^T\right] \tag{4.18}$$

where $\lambda > 0$ is the regularisation parameter, while $\mathbf{A}_t$ and $\mathbf{B}_t$ are the Laplacian matrices, given by:

$$\mathbf{A}_t = \text{diag}(\mathbf{A}_t' \times \mathbf{1}_{n\times 1}) - \mathbf{A}_t' \tag{4.19}$$

$$\mathbf{B}_t = \text{diag}(\mathbf{B}_t' \times \mathbf{1}_{n\times 1}) - \mathbf{B}_t' \tag{4.20}$$

where the diag$(\cdot)$ function returns a vector corresponding to the matrix diagonal for a given input matrix or a corresponding diagonal matrix for an input vector. $\mathbf{A}_t' = [a_{ij}'^t]$ and $\mathbf{B}_t' = [b_{ij}'^t]$ are the proximity matrices corresponding to the weights in (4.16) and (4.15) respectively, as:

$$a_{ij}'^t = \text{sign}(\gamma_{ij}^t)[\text{sign}(\gamma_{ij}^t) - 1]\gamma_{ij}^t f_{ij} \tag{4.21}$$

$$b_{ij}'^t = \text{sign}(\gamma_{ij}^t)[\text{sign}(\gamma_{ij}^t) + 1]\gamma_{ij}^t f_{ij} \tag{4.22}$$

A flow digram summarising the steps of calculating the Laplacian matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ is presented in **Algorithm 1**. $\mathbf{1}_{n\times 1}$ is a $n \times 1$ vector with its all elements to be 1.

Based on (4.17) or (4.18), there are c problems of eigen decomposition that need to be solved for a given dataset of c classes. For practical applications where time consuming is critical, an alternative trace optimisation problem is proposed to reduce the computational complexity, as:

$$\underset{\mathbf{V}\mathbf{V}^T=\mathbf{I}_{k\times k}}{\arg\max}\ \text{trace}\left[\mathbf{V}\mathbf{X}^T(\mathbf{A}_{total}-\lambda\mathbf{B}_{total})\mathbf{X}\mathbf{V}^T\right] \tag{4.23}$$

or

$$\underset{\mathbf{V}(\mathbf{X}^T\mathbf{B}_{total}\mathbf{X}+\lambda\mathbf{I}_{d\times d})\mathbf{V}^T=\mathbf{I}_{b\times b}}{\arg\max}\ \text{trace}\left[\mathbf{V}\mathbf{X}^T\mathbf{A}_{total}\mathbf{X}\mathbf{V}^T\right] \tag{4.24}$$

where $\lambda > 0$ is a regularization parameter, while $\mathbf{A}_{total}$ and $\mathbf{B}_{total}$ are the sums of the Laplacian matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ $(t = 1, \ldots, c)$, given by:

$$\mathbf{A}_{total} = \mathbf{A}_1 + \mathbf{A}_2 + \cdots + \mathbf{A}_c \tag{4.25}$$

Algorithm 1 Description of the main steps of the proposed method to calculate the Laplacian matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ $(t = 1, \cdots, c)$.

Input: The training set $\mathbf{X}$, and the label information $\mathbf{Y}$
Output: The Laplacian matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ $(t = 1, \cdots, c)$
Initialisation
1: (Optionally) Apply the data pre-processing technique PCA (listed in appendix B.1) to the training set $\mathbf{X}$
2: Compute the feature based weight matrix $\mathbf{F}$ using the Gaussian Weight technique (listed in appendix A)
LOOP Process
3: **for** $t = 1$ to c **do**
4: Obtain the indicator for weight pairwise constraint $\mathbf{\Gamma}^t = [\gamma_{ij}^t]$ using (4.14)
5: Compute the proximity matrices $\mathbf{A}_t'$ and $\mathbf{B}_t'$ using (4.21) and (4.22)
6: Obtain the Laplacian matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ using (4.19) and (4.20)
7: **end for**
8: **return** The Laplacian matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ $(t = 1, \cdots, c)$

$$\mathbf{B}_{total} = \mathbf{B}_1 + \mathbf{B}_2 + \cdots + \mathbf{B}_c \tag{4.26}$$

The optimal projection matrix $\mathbf{V}^*$ can be obtained by solving the standard eigen problem (4.23) or the generalized eigen problem (4.24). Then the optimal embedded data samples of both the training set $\mathbf{X} = [x_{ij}]$ and the testing set $\widetilde{\mathbf{X}} = [\widetilde{x}_{ij}]$ can be computed as $\mathbf{Z} = \mathbf{X}\mathbf{V}^*$ and $\widetilde{\mathbf{Z}} = \widetilde{\mathbf{X}}\mathbf{V}^*$.

4.3 Framework Evaluation

To evaluate the proposed binary data embedding framework, it was compared to seven popular techniques for embedding learning: three unsupervised ones, including PCA [42], OLPP [45] and ONPP [45]; four supervised ones, including FDA [50], LFDA [57], DNE [54] and OLPP-R [60]. All of them haven been descried in chapter 2. The main conceptual difference between the proposed method and the existing ones is that it is able to identify one or more unique features for each class individually. All experiments were conducted using MATLAB R2014b on a machine with 3.2-GHz Intel Core i5 CPU and 16-GB DDR3 RAM running the Ubuntu operating system.

4.3.1 Datasets

To demonstrate the generalisation and applicability of the proposed binary data embedding framework to relevant areas, popular face databases and text datasets from the literature were used.

Face Databases

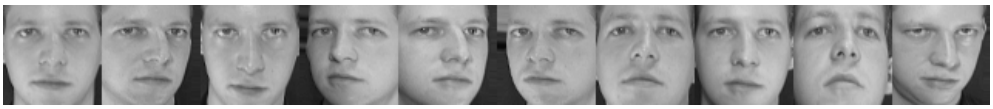
Four popular face databases for benchmarking multiclass classification were used: the Yale Face Database (Yale) [55], the ORL Database of Faces (ORL) [276], the PIE

Database, CMU (PIE) [277] and the Extended Yale Face Database B (YaleB) [278, 279]. Yale [55] contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration. ORL [276] contains 400 different images of 40 distinct subjects. Each has 10 different images, with the images taken at different times, varying the lighting, facial expressions and facial details. PIE [277] is a database of $41,368$ images of 68 people, each person has 13 different poses, 43 different illumination conditions, and with 4 different expressions. In this experiment, to compare with the other three face databases, which only use near frontal pose images, only 5 near frontal poses and all images under different illuminations and expressions were used, which resulted about 170 images for each individual. YaleB [278,279] contains $16,128$ images of 38 human subjects with 9 poses and 64 illumination conditions. In the experiment, only the cropped images were used, which resulted in 38 individuals and around 64 near frontal images under different illuminations per individual. All data samples (i.e. face images) of the four face databases were resized to 64×64 pixels, and then the pixels were converted into a column vector of dimension $4,096$, which resulted a feature size of $4,096$. To show the differences between each data samples in the same class, all the data samples in a single example class from each of the four face databases (i.e. Yale, ORL, PIE and YaleB) are displayed in Figure 4.1a, 4.1b, 4.1c and 4.1d, respectively.

Text Datasets

Two popular text datasets were also used: the 20 Newsgroups Dataset (20News) [280] and the Reuters Corpus [281]. 20News [280] is a collection of approximately $20,000$ newsgroup documents, partitioned nearly evenly across 20 different newsgroups. The "by date" version was used in this experiment, which contained $18,846$ documents [280]. Then, after stemming and removing stop words, there were $26,214$ distinct terms (i.e. feature size). All terms were listed as a row, and each document (i.e. data sample) was assigned a binary value of 1 if it contained this word, 0 otherwise. Reuters [281] contains $21,578$ documents in 135 categories. In this experiment, the documents with multiple category labels were discarded, and also only the first 11 largest categories were kept, which left $7,372$ documents. The same method for assigning binary values

(a) Yale



(b) ORL

(c) PIE

(d) YaleB

Figure 4.1: All the data samples in a single example class from each of the four face databases: Yale, ORL, PIE, and YaleB.

was used, and resulted a feature size of $18,933$.

4.3.2 Experimental Design

A Hold-out scheme was adapted in all experiments on the six different datasets for model assessment. Data samples from each of the six datasets were divided into three independent partitions, for the purpose of training, validation and testing. The number of data samples for training was kept relatively small (i.e. 20% for training, while 40% was used for validation and testing), in order to show that the proposed method can generate robust embeddings for classification, even when there is only a small number of data samples available for training. 200 random permutations were used, which resulted in 200 different groups of data samples for training, validation and testing.

The details of characteristics and partitioning for each of the six datasets are summarised in Table 4.1. In addition, to visually analyse the original class structures, the proximity matrices were computed for all six datasets, based on the Gaussian Weight (eq. (A.5)). The resultant proximity matrices are displayed in Figure 4.2 for the four face databases and two text datasets, respectively. The original features of the entire testing sets were used, and additionally, data samples are ordered to ensure that

Table 4.1: Characteristics and partitioning of the datasets.

Dataset	Classes	Features	Training	Validation	Testing
Yale	15	4,096	33	66	66
ORL	40	4,096	80	160	160
PIE	68	4,096	2310	4622	4622
YaleB	38	4,096	482	966	966
20News	20	26,214	3770	7538	7538
Reuters	11	18,933	1659	3317	3317

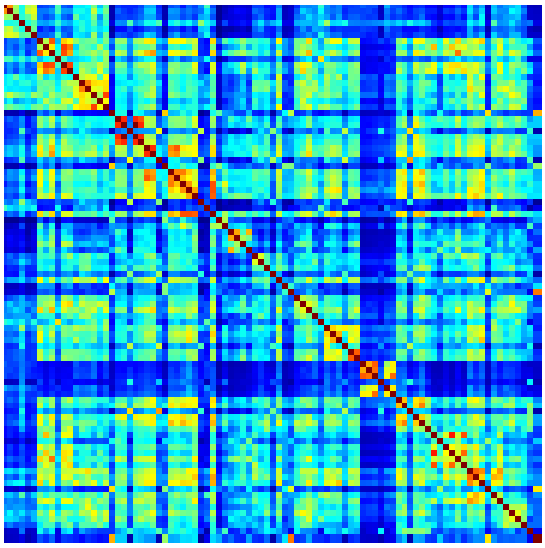
Table 4.2: The smallest and largest ratios of the number of data samples in a target class to that in the corresponding remaining classes, for each derived problem of binary classification in the six used datasets.

	ratio of no. of data samples in binary classification (%)	
	smallest ratio	largest ratio
Yale	7.14	7.14
ORL	2.56	2.56
PIE	1.42	1.47
YaleB	2.44	2.65
20News	3.33	5.30
Reuters	1.18	50.37

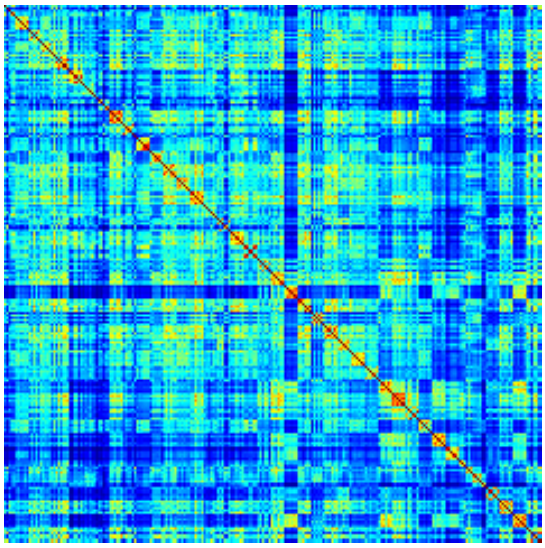
those of the intraclass appear consecutively when plotted. From the proximity matrices shown in Figure 4.2a-4.2f, it can be seen that the original features of Yale and ORL show some class separability, but the class structures are not separated distinctly. This implies the existence of ambiguous interclass data samples. The situation is worse in the cases of PIE, YaleB and Reuters, and it is even more challenging in the case of 20News.

For all the six different datasets used for benchmarking, since each of them corresponds to a multiclass classification problem while the proposed method utilises the idea of OVA, then there is a various number of binary classification problems derived for each of the six used datasets. To show that there are indeed many imbalanced cases in the derived binary classification problems (i.e. the number of data samples in the two classes for classification is imbalanced), the ratio of the number of data samples in a target class to that in the corresponding remaining classes was calculated, for each of the derived binary classification problems. Subsequently, for each of the six different datasets, the smallest and largest ratios among the calculated ratios are shown in Table 4.2. From Table 4.2, it can be seen that for all the six used datasets, there is always an imbalanced case in each binary classification problem derived by the proposed method.

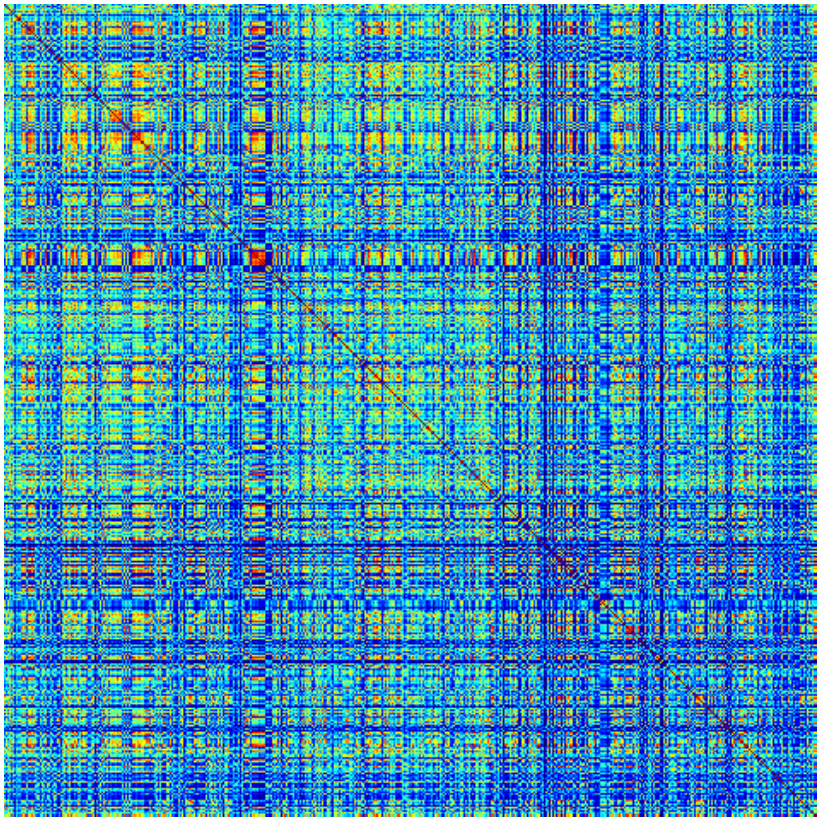
To demonstrate the results obtained in the embedding computation stage, a simple classifier is sufficient for identification and classification. A simple nonparametric nonlinear classifier, the 1-NN, was selected for the final stage of identification and classification. The parameters for the proposed method and all comparison techniques were tuned by a grid search. The process of parameter optimisation for all experiments and all methods were performed using the validation sets, while the final assessments were



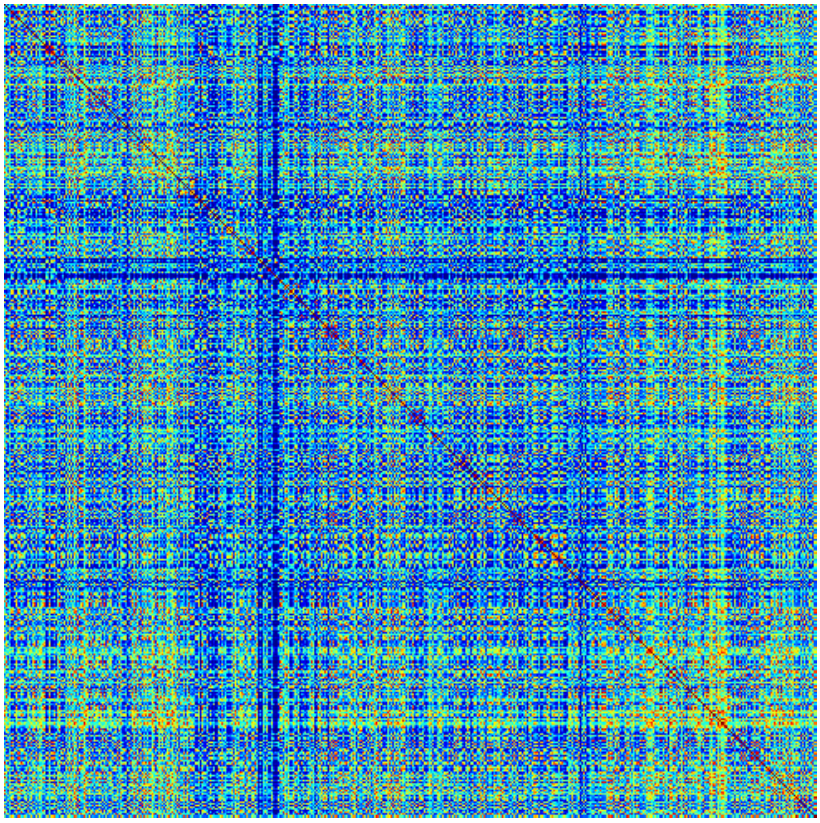
(a) Yale



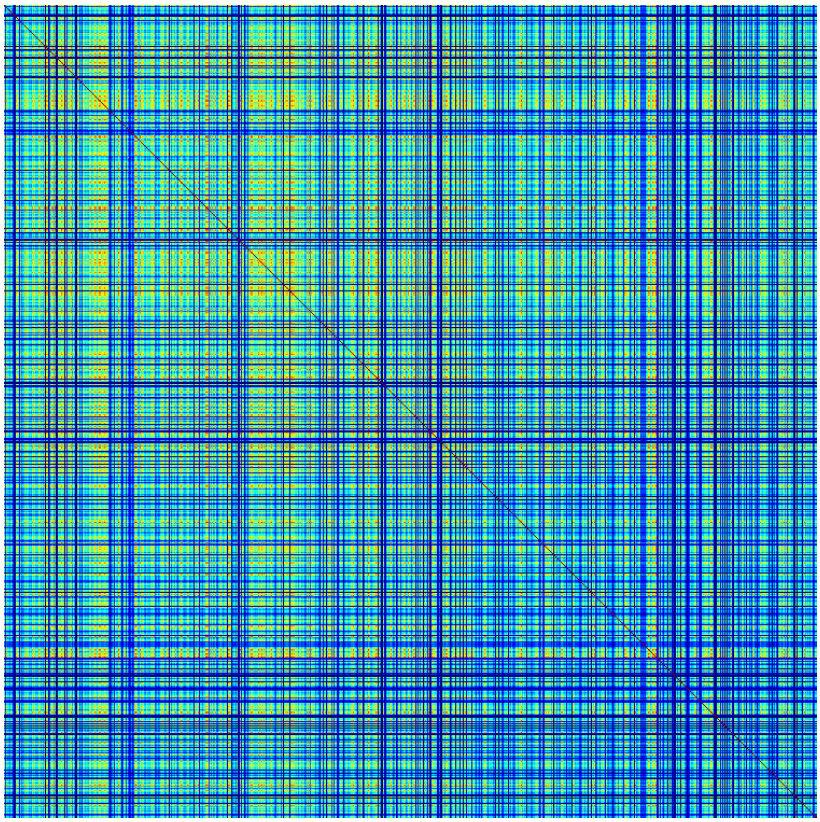
(b) ORL



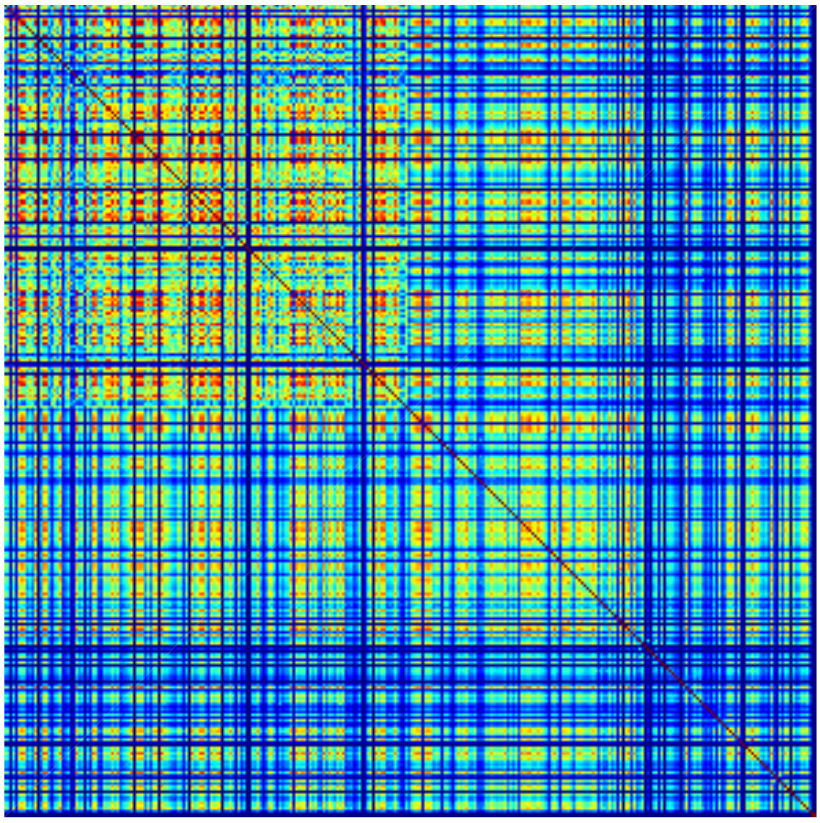
(c) PIE



(d) YaleB



(e) 20News



(f) Reuters

Figure 4.2: Comparison of the class structures based on the proximity matrices computed from the original data samples (the entire training sets were used).

conducted using the testing sets [14]. To assess the performance, accuracy rate was selected, which is the most commonly used metric [282, 283]. It refers to the number of correctly classified data samples (i.e. successful hits) relative to the total number of ones for classification [269].

Sensitivity Analysis

A sensitivity analysis was performed for two parameters, i.e. k (the k-NN for the local neighbour information), and σ (the parameter that controls the bandwidth of the Gaussian kernel). The sensitivity analysis was conducted on YaleB [278, 279]. To simplify this analysis, b was kept to be equal to the total number of classes (i.e. 38). k was varied from 1 to 19, with a constant step of 2 (i.e. $1, 3, \ldots, 19$). The variation of σ is more generic, because the value of $\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$ changes when different testing sets are used. As a result, it was set to be adaptive to the median value of $\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$, with the use of the corresponding InterQuartile Range (IQR): from median($\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$)$-2\times$IQR to median($\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$) $+ 2 \times$ IQR, i.e.:

$$\sigma = \text{median}(\ \|\mathbf{x}_i - \mathbf{x}_j\|_2^2) + c_\sigma \times \text{IQR} \tag{4.27}$$

where c_σ was varied from -2 to 2, with a constant step of 0.5 (i.e. $-2, -1.5, \ldots, 2$).

The details of the sensitivity analysis of the performance of the proposed method on YaleB is shown in Figure 4.3, with the use of error rate (i.e. $1 -$ accuracy rate). From Figure 4.3, it can be seen that the proposed method achieved the best performance (i.e. a error rate of 6.33%) when k and σ were set to be 19 and median($\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$)+1.5×IQR respectively, and the worst performance (i.e. a error rate of 7.66%) when k and σ were set to be 1 and median($\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$) $- 1.5 \times$ IQR respectively. The difference rate is only 1.33%, which is fairly small. As a result, the performance of the proposed method on YaleB is not sensitive to the variations of its parameters k and σ. However, in all experiments here, the parameters of all methods were set to the optimal values by the process of model selection, with the use of the validation sets.

4.3.3 Benchmark Evaluation

To evaluate the performance of the proposed method, it was compared against seven existing techniques, using the six datasets listed in Table 4.1. Since the data samples in each of the datasets have a very high dimensionality and these embedding learning methods are all based on an eigen decomposition of a $d \times d$ matrix, dimension reduction becomes necessary in the pre-processing stage to reduce the following computational complexity. By using the data pre-processing technique PCA (eq. (B.8)), data samples can be transformed from the original high dimensional feature space to a much lower one [263]. After employing the data pre-processing technique PCA, the resultant dimensionality r was determined so that over 99% variance was kept in the pre-processed data samples. All of the methods used for comparison and the proposed one were implemented on these pre-processed data samples. The final performance of each method was obtained by averaging over 200 different experiment results. The method performances on the four face databases and two text datasets are summarised within Table 4.3 and 4.4, respectively.

It can be seen from Table 4.3 that, the results obtained from the pre-processed data samples can achieve almost the same accuracy rate, compared to those obtained from the original data samples. This demonstrates that the data pre-processing technique PCA does not remove any significant structural information from the original data samples, but does reduce the computational complexity. The proposed algorithm either outperforms the seven existing methods or achieves the same performance for all six datasets, except in the case of Yale, where FDA achieved a better average accuracy rate than the proposed method. However, the difference is very small, it is only 0.13%, which only results a difference rate of 0.09 data sample, as the size of Yale is fairly small.

For the six different datasets, to approximately analyse whether there exist some empirical thresholds on the minimum number of training data samples required to

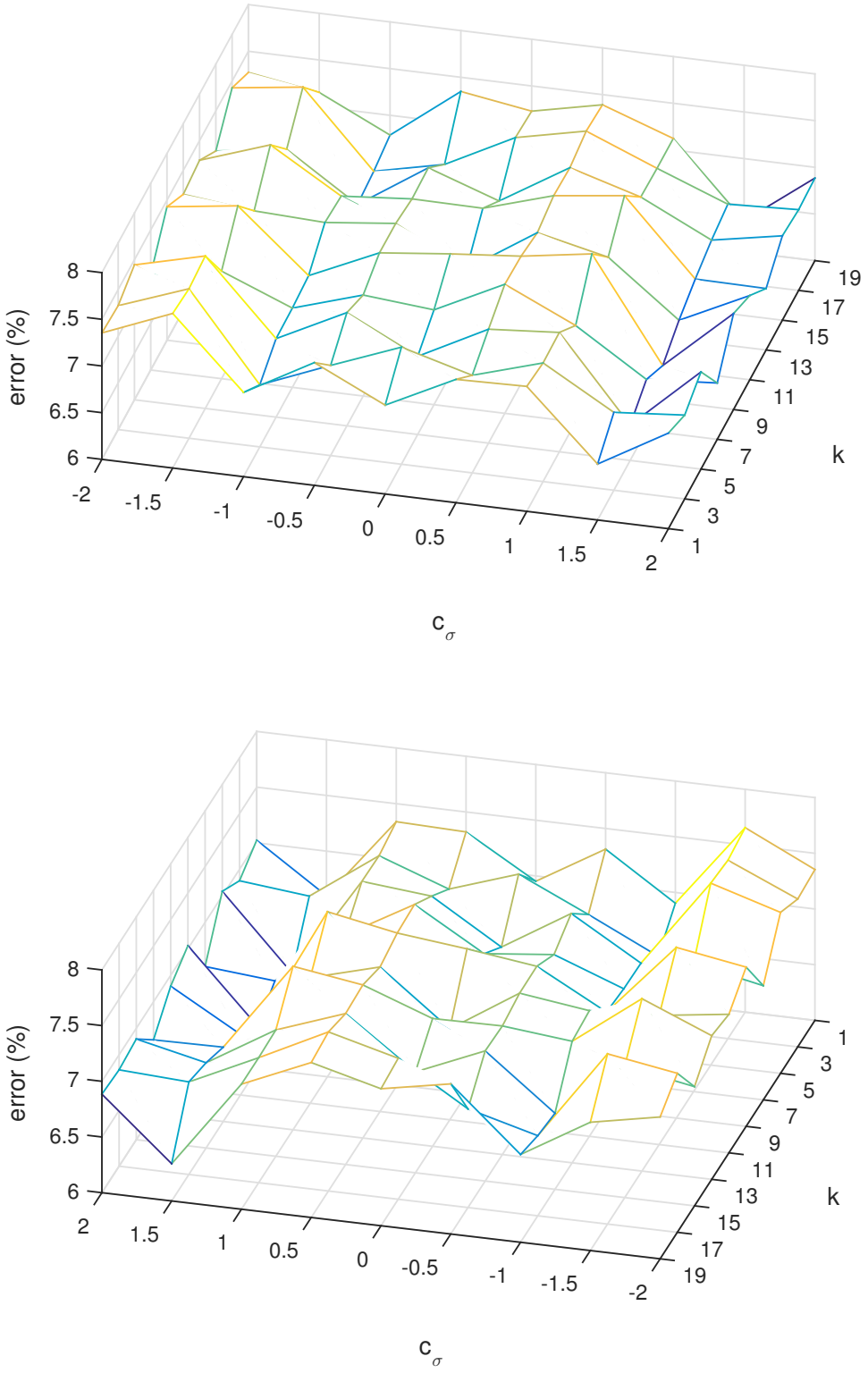
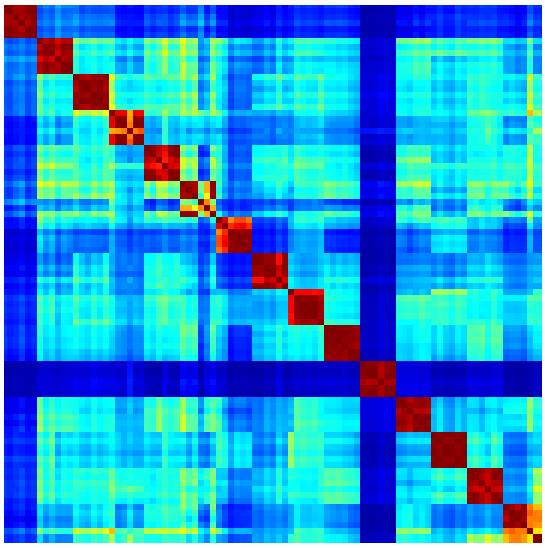
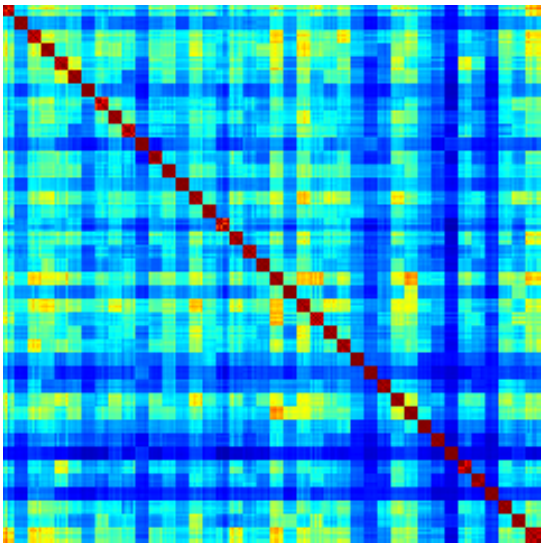


Figure 4.3: Sensitivity analysis of the performance of the proposed method to the variations of the parameters k and σ (in terms of c_σ), on YaleB.

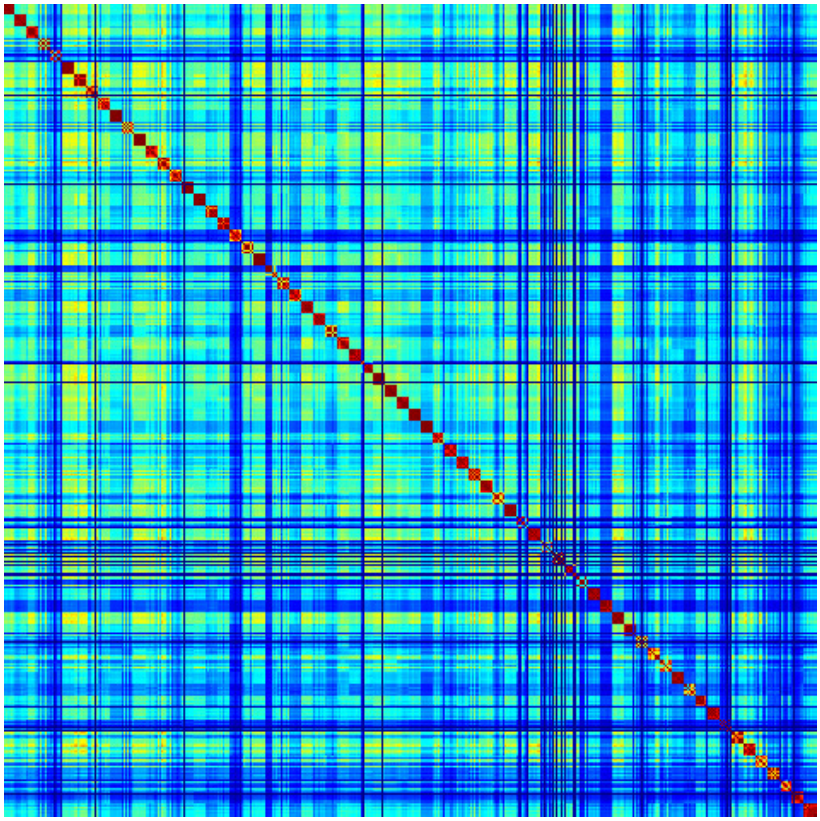
achieve reasonable performances, the proposed method was tested with varying the number of training data samples. For the datasets Yale and ORL, the number of training data samples was varied from 20% to 60% with the step number to be 10%, since the size of the data samples in each class is fairly small. While for the remaining four large datasets, the number of training data samples was varied from 5% to 40% with the step number to be 5%. Meanwhile, the number of data samples for validation



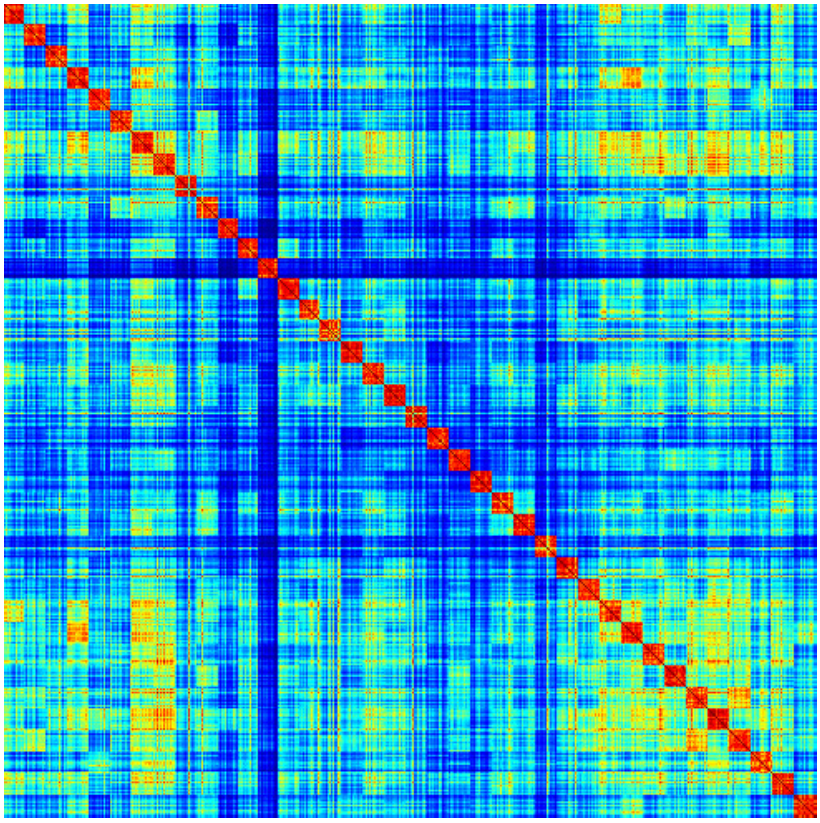
(a) Yale



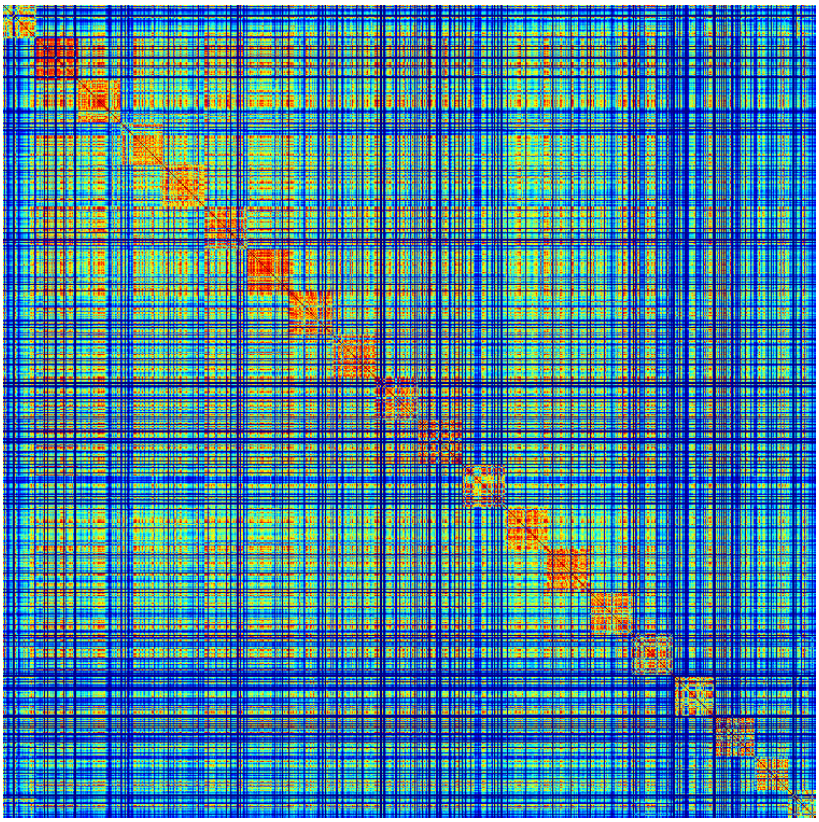
(b) ORL



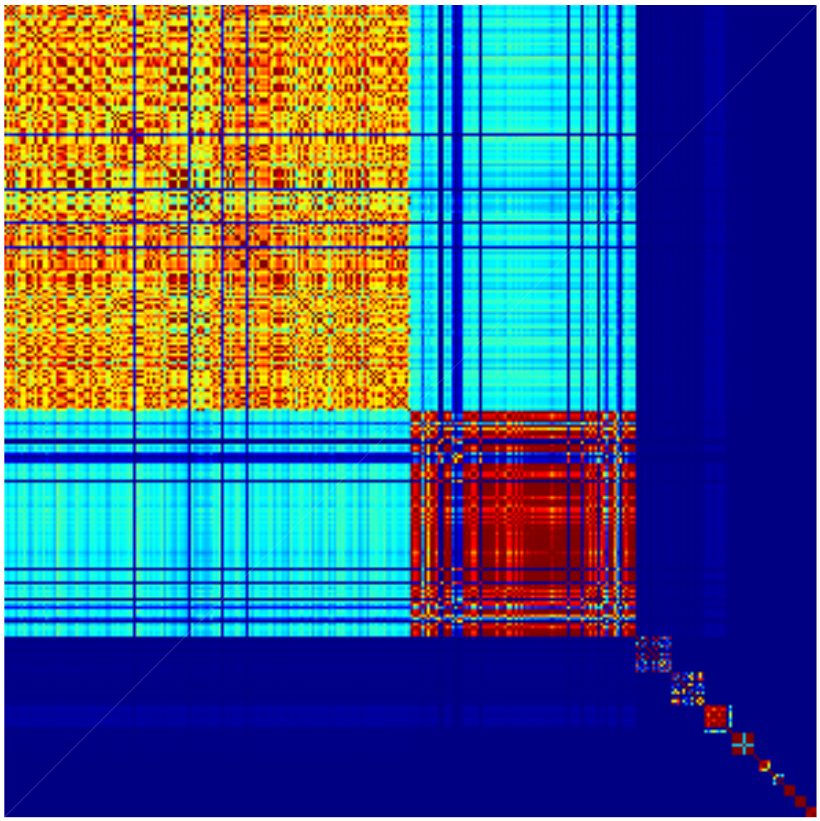
(c) PIE



(d) YaleB



(e) 20News



(f) Reuters

Figure 4.4: Comparison of the class structures based on the proximity matrices computed from the embedded data samples generated by the proposed method (the entire training sets were used).

Table 4.3: Performance comparison in terms of accuracy rate of classification and identification, using the test partitions of the four face databases. The final dimensionalities b of the different embeddings are shown in parentheses.

	Yale (%)	ORL (%)	PIE (%)	YaleB (%)
Original	70.86(4096.0)	90.59(4096.0)	92.45(4096.0)	66.82(4096.0)
Pre-processed	70.38(77.4)	90.42(176.2)	92.29(554.8)	65.06(177.6)
PCA	69.94(54.1)	90.38(123.2)	92.25(518.5)	64.72(163.6)
OLPP	69.81(72.1)	93.13(167.1)	94.52(526.8)	88.24(150.0)
ONPP	69.23(68.8)	92.98(166.2)	94.93(518.6)	86.96(147.2)
FDA	**87.88**(13.7)	95.46(38.3)	96.97(61.3)	93.55(37.9)
LFDA	85.79(18.0)	95.38(38.8)	96.37(63.7)	93.27(48.4)
DNE	80.54(15.8)	94.53(38.2)	96.57(97.8)	91.32(48.1)
OLPP-R	82.93(14.3)	**95.49**(34.3)	97.08(59.9)	93.35(46.4)
Proposed	87.75(15.3)	**95.49**(39.5)	**97.40**(62.1)	**93.89**(39.7)

Table 4.4: Performance comparison in terms of accuracy rate of classification and identification, using the test partitions of the two text datasets. The final dimensionalities b of the different embeddings are shown in parentheses.

	20News (%)	Reuters (%)
Original	62.21(26214.0)	85.95(18933.0)
Pre-processed	63.25(1986.9)	87.95(2555.3)
PCA	62.77(1390.6)	88.86(1788.7)
OLPP	58.05(1887.8)	77.27(2120.7)
ONPP	63.95(1390.6)	83.33(2427.4)
FDA	76.75(19.0)	91.43(35.0)
LFDA	68.34(113.1)	90.18(199.4)
DNE	76.62(312.1)	93.22(173.8)
OLPP-R	63.71(213.1)	91.47(10.0)
Proposed	**76.86**(19.0)	**93.37**(135.6)

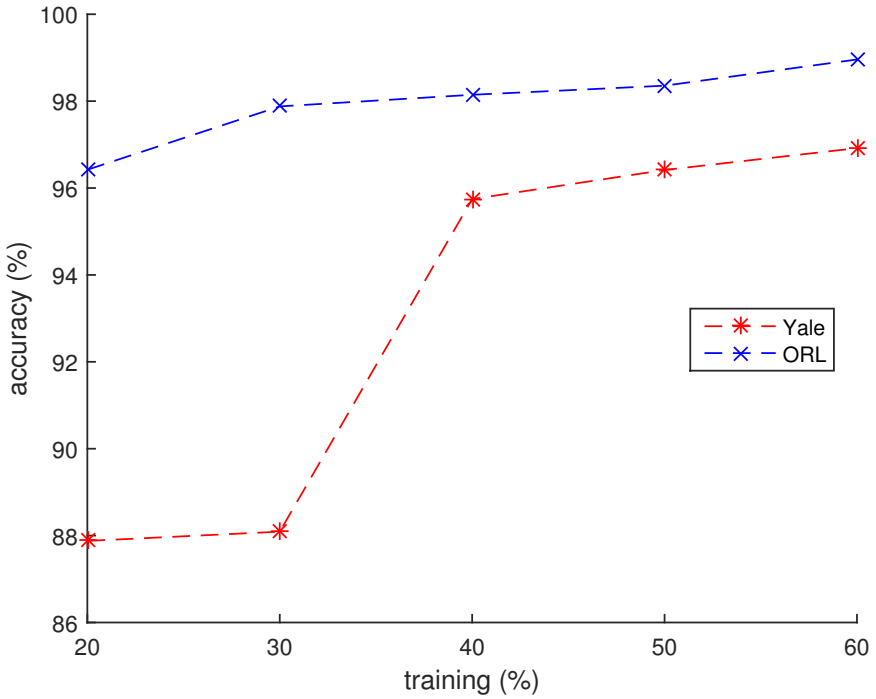
and testing were always kept to be approximately equal. The corresponding results for the the datasets Yale and ORL, the datasets PIE and YaleB, as well as the datasets 20News and Reuters are shown in Figure 4.5a, Figure 4.5b, and Figure 4.5c respectively. From Figure 4.5, it can be seen that for PIE, there only requires the number of training data samples to be 15% so that it can achieve a reliable performance, which means that there is a sufficient number of similar data sample available in each class; while for ORL, YaleB, 20News and Reuters, the minimum number of training data samples required to achieve reasonable performances are 20%, 20%, 25% and 20%, respectively; finally, for Yale, in order to get a desirable result, it needs the number of training data samples to be at least 40%, which implies that the data samples in each class are very dissimilarly to each other.

In order to demonstrate the improved class separability using the proposed method, the same training sets for computing proximity matrices shown in Figure 4.2 were used to compute the embedded data samples generated by the proposed method. Then, the corresponding six proximity matrices were displayed in Figure 4.4. Comparing Figure 4.2 and Figure 4.4, it can be seen that the proximity matrices obtained from the embedded data samples generated by the proposed method possesses many more distinct class structures than those computed from the original ones. In other words, in Figure 4.4 the data samples from the same class are tending to group into blocks in the diagonal direction (as data samples are ordered to ensure those of intraclass appear together when plotted). It means that, in the embedded space processed by the proposed method, the data samples of intraclass are grouped together, and additionally, there are suitable separation distances between those of interclass.

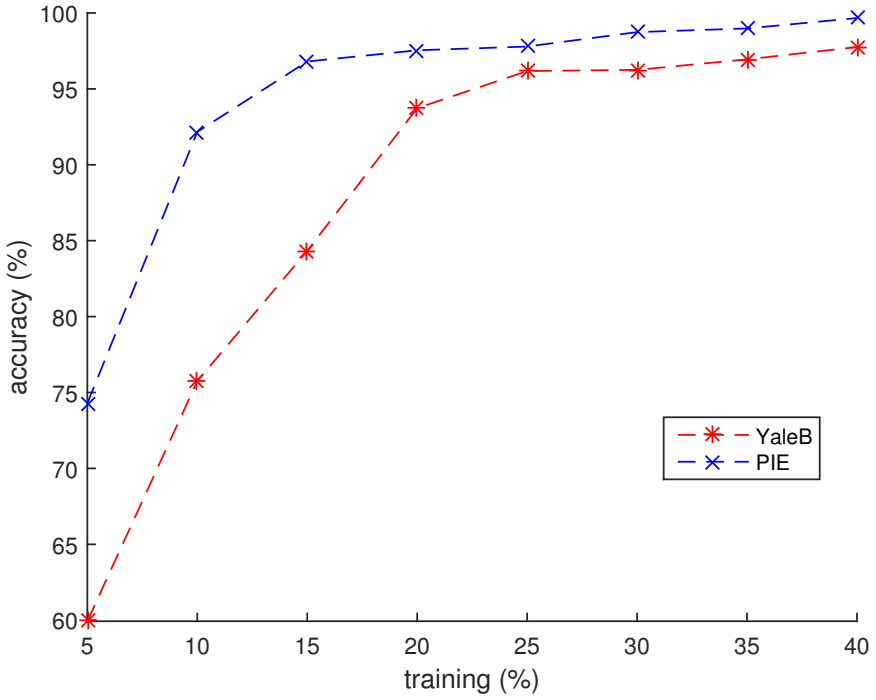
To numerically compare the proximity matrices computed from the original data samples (shown in Figure 4.2) and those computed from the embedded data samples generated by the proposed method (shown in Figure 4.4), both 1-NN classifier and k-means clustering were employed, and the results are shown in Table 4.5. From Table 4.5, it can be seen that for all the six datasets, as well as both methods 1-NN and k-means, the performances on the embedded data samples generated by the proposed method are much enhanced compared those on the original data samples, which further demonstrates the power of the proposed method in improving class separability.

Computational Cost Analysis

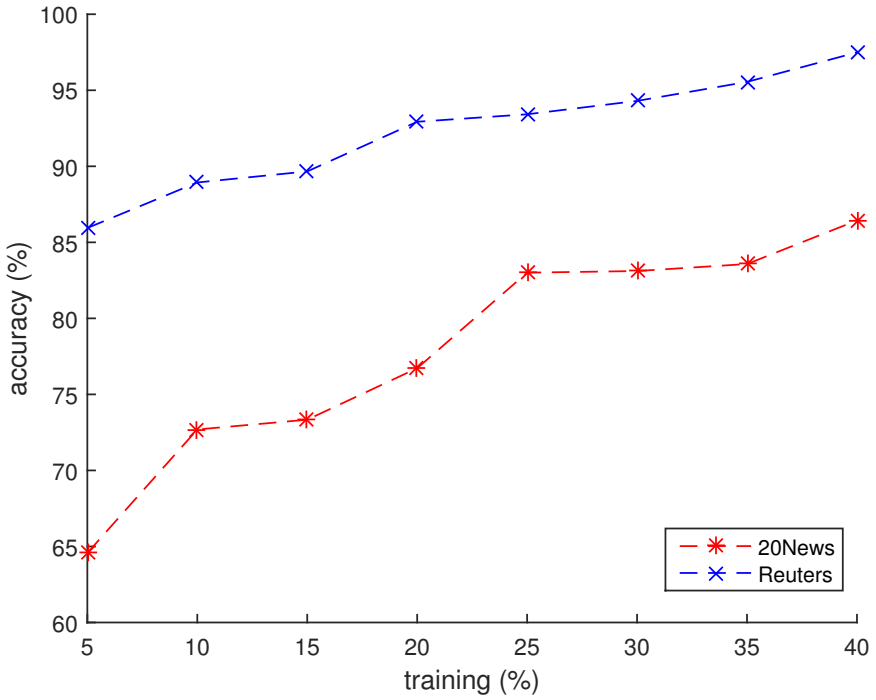
Compared to the traditional technique for binary classification applied to multiclass classification problem, the proposed method outperforms with respect to computational complexity. Rather than using a very complex search algorithm to decide the label for a testing data sample, the proposed method incorporates the idea of OVA in the manifold embedding. This provides better performance, while reducing the original computational complexity associated with the binary decomposition approach to multiclass classification.



(a) Yale & ORL



(b) PIE & YaleB



(c) 20News & Reuters

Figure 4.5: Performance of the proposed method, in terms of accuracy rate of classification and identification, on the six different datasets with varying the number of data samples for training.

Table 4.5: Numerical comparison of the proximity matrices computed from the original data samples and those computed from the embedded data samples generated by the proposed method, in terms of accuracy rate of classification and identification, using 1-NN classifier and k-means clustering. The number of clusters k for the six different datasets are shown in parentheses.

		Original (%)	Embedded (%)
Yale	1-NN	50.00	100.00
	k-means ($k = 15$)	50.00	100.00
ORL	1-NN	55.00	100.00
	k-means ($k = 40$)	66.25	100.00
PIE	1-NN	66.03	98.31
	k-means ($k = 68$)	29.22	81.09
YaleB	1-NN	46.94	97.14
	k-means ($k = 38$)	24.69	89.49
20News	1-NN	44.56	81.03
	k-means ($k = 20$)	25.28	70.88
Reuters	1-NN	78.70	95.32
	k-means ($k = 11$)	36.97	83.65

4.4 Summary

In this chapter, a novel manifold embedding method for the automated processing of large varied datasets has been proposed. The proposed method consists of two stages: preprocessing and embedding computation, where the embedding computation stage can be seen as a generalised process of the Fishers criterion [50], while additionally somehow incorporating the local neighbour information. The proposed method is based on binary classification, and it aims to generate binary decision processes, such that a low-dimensional embedding is constructed while the unique characteristics of each individual class can also be determined. To address the issue of an imbalanced number of data samples that is usually associated with the binary decomposition approach to multiclass classification (i.e. the number of data samples from a target class is relatively small, compared to that of the remaining data samples), instead of selecting data samples from the remaining ones to achieve a balance, the proposed method employs a weighted pairwise constraint indicator. The weighted pairwise constraint indicator is designed so that the contributions of the data samples from a target class and those from the remaining classes are balanced, additionally the relative positions between the important data samples from either the same class (intraclass) or different classes (interclass) are better controlled. The effectiveness of the proposed method has been evaluated through comparison with seven existing techniques for embedding

learning (both unsupervised and supervised), using four established databases of faces, consisting of various poses, lighting conditions and facial expressions, as well as two standard text datasets. It has been shown that the proposed method can outperform these seven existing techniques when there is only a small number of data samples available for training.

Chapter 5

Image Reconstruction by Fusing Low Bit Depth Imagery

This chapter is mainly based on the published work [30].

5.1 Motivation

In recent years, the use of Unmanned Aerial Vehicles (UAVs) for various civilian purposes has been growing rapidly, and a number of relevant applications have been developed [284–286]. For example, delivering medical support, search and rescue operations, border patrol missions, surveillance, land surveying, and crowd monitoring. Usually, UAVs are equipped with sensors (e.g. visible band or infrared cameras) capturing images, so that aerial images of the scene can be obtained and used for further information retrieval [287].

For law enforcement and military applications (e.g. border patrol missions, surveillance, and crowd monitoring), a secure data link is often required between the UAV and a remote operator for transmitting data [288]. The data link will always have a limited data rate, which is limited further by the need for encryption and other secure communications overheads [29]. However, advances in digital camera and video technology allows higher resolution images and increasing frame rates, which aggravates the problems associated with data transmission. The traditional approach to this problem is data compression [289], where image data is compressed via a complex algorithm before transmission and later decompressed at the receiver. Usually, the compression process is quite computationally expensive. In addition, operational and cost factors often mean that UAVs used in law enforcement operations are relatively small (compared to military systems) and have limited computational resources [287]. To reduce the usage of on-board computational resources, an alternative approach can be employed by using low bit depth imagery, which only transmits the first few Most Significant Bits (MSBs) of image data. Some work has been done to investigate the use of low bit depth imagery [29, 33, 290–292]. This approach assumes the most im-

portant information of each image (e.g. the main structure of the scene) is contained in the MSBs. However, low bit depth imagery can discard image details within the Least Significant Bits (LSBs), which could cause problems in applications which focus on image details, such as the identification or recognition of objects. Work has been done to address this problem [29], and it has been found that the discarded details may be reconstructed by fusing a sequence of related low bit depth images if there is sufficient noise in the images, when the temporal sequence has been aligned by image registration.

5.2 Image Reconstruction Process

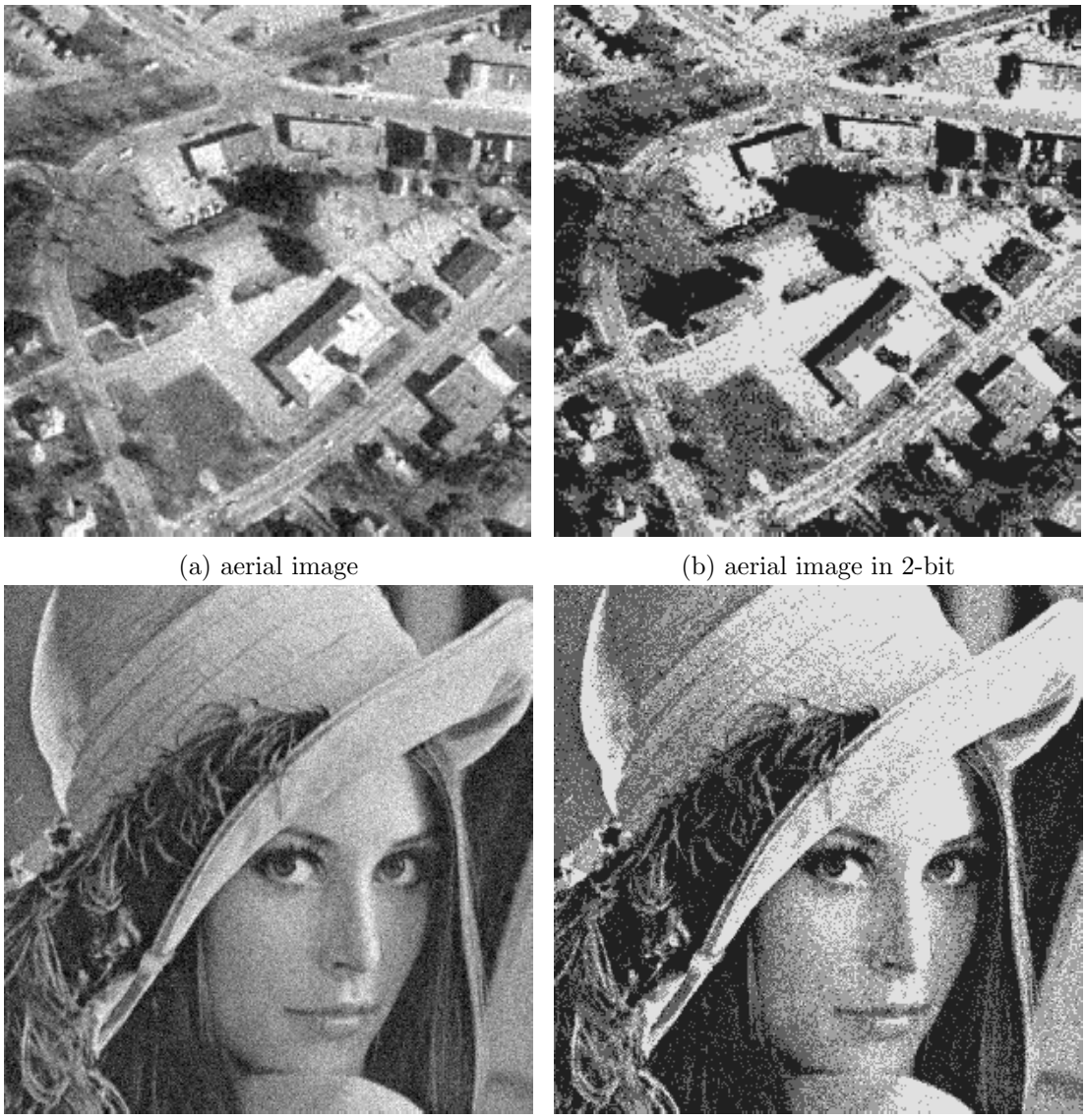
The image reconstruction process consists of the following three steps: generating low bit depth images, registering generated low bit depth images, and fusing aligned low bit depth images.

Generation of Low Bit Depth Images

A sequence of 2-bit images are considered low bit depth imagery. To equally divide the pixel intensity range (i.e. 0 to 255) into four sub-ranges for 2-bit image encoding (i.e. {00b, 01b, 10b, 11b}), three quantisation thresholds are set, respectively at 64, 128 and 192. It results of four equal pixel intensity sub-ranges: 0 to 63, 64 to 127, 128 to 191, and 192 to 255. In order to minimise the quantisation error, on reconstruction, the grey-levels after quantisation are chosen to be the mid point of the sub-ranges, i.e. 32, 96, 160 and 224. Low bit depth images can be sensitive to noise fluctuating near the thresholds, causing large intensity differences on reconstruction, although when averaged this can contain useful information [29]. The conversion from a 8-bit depth image to a 2-bit depth image can be easily accomplished by: preserving the 2 MSBs of the full-bit depth image image, whilst always setting the third MSB to one and the remaining 5 bits to zero. Examples of two full bit depth images and the corresponding low bit depth images in 2-bit are shown in Figure 5.1.

Registration of Low Bit Depth Images

Mathematically, a projective transformation (up to 8 degrees of freedom) is required to describe the geometrical alignment between two images of the same 2D scene. However, the projective transformation is likely to be influenced by small errors, and any small distortion will spread after a number of images [287]. To overcome this problem, a similarity transformation (up to 4 degrees of freedom) is used instead of a projective transformation for images captured by small or micro-scale UAVs. A similarity transformation consists of a uniform scaling, a rotation, and a translation. Under a similarity distortion, angles between lines are preserved, which excludes more complex



(a) aerial image (b) aerial image in 2-bit

(c) Lena (d) Lena in 2-bit

Figure 5.1: Examples of two 8-bit depth images and the corresponding 2-bit depth images: Gaussian noise is present in the 8-bit depth images, and the image structures are preserved in the 2-bit depth images.

transformations, such as shearing. As long as the UAV holds an approximate nadir view (looking downwards), using a similarity transformation to approximate the geometrical alignment between two images can lead to a less deformation of the resulting images [287, 293]. Since only a similarity transformation needs to be estimated, the Phase Correlation (Fourier) method is employed here. Phase Correlation (PC) can achieve a low computational complexity, because it can be computed efficiently for images using the Fast Fourier Transform (FFT) [17, 152].

Fusion of Low Bit Depth Images

The image details of the target scene discarded due to quantisation may be reconstructed by fusing a number of low bit depth images of the target area, which have been properly aligned [29]. The resultant fused image is easier to analyse or interpret than any individual source image. Because the image contrast changes from image to image, images of the same target scene taken at different times, from different viewpoints, and/or by different sensors are more likely to have a large variations in grey levels for the same pixel location. These crossings of the quantisation thresholds contains information about the original image, which can be recovered by averaging [29]. After quantisation, image details would initially appear to have vanished. However, once the images have been aligned by the image registration process, the contribution from the differences in spatial and intensity to the recovery of finer image details is beyond that one would expect from a single image.

Here, the 2-bit images are combined together by using a simple weighted sum. The weight associated with each 2-bit image depends on the quality of image registration result, which can be measured based on the Normalised Cross Correlation (NCC) between the reference and registered sensed images I_R and $\tilde{I}_S$, respectively as:

$$NCC = \frac{\sum_x \sum_y (I_R(x,y) - \mu_R)(\tilde{I}_S(x,y) - \mu_S)}{\sqrt{\sum_x \sum_y (I_R(x,y) - \mu_R)^2 (\tilde{I}_S(x,y) - \mu_S)^2}} \tag{5.1}$$

where μ_R and μ_S are the mean images of the corresponding images, and they are computed according to:

$$\mu_R = \frac{1}{N} \sum_x \sum_y I_R(x,y) \tag{5.2}$$

$$\mu_S = \frac{1}{N} \sum_x \sum_y \tilde{I}_S(x,y) \tag{5.3}$$

where N is the number of pixels in the image. Also the NCC result can be used to indicate poorly registered images, by setting a certain threshold. Any poorly aligned image should be removed from the fusion process, since it is likely to degrade the quality of the final fused result [29].

Given a number of m aligned images $I_1, I_2, \ldots, I_m$, the pixel intensity value of the resultant fused image I_0 at the pixel location (x, y) can be computed by:

$$I_0 = \sum_{i=1}^{m} w_i I_i \tag{5.4}$$

where the weight w_i can be obtained as:

$$w_i = \begin{cases} NCC_i, & \text{if } I_i \text{ has a valid pixel intensity value} \\ & \text{at the pixel location } (x,y) \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

where NCC_i represents the NCC result between the aligned image I_i $(i = 1, \ldots, m)$ and its reference image. Additionally, the weights $\{w_i\}_{i=1}^{m}$ are normalised such that

$$\sum_{i=1}^{m} w_i = 1 \tag{5.6}$$

5.3 Experimental Evaluation

Generation of Low Bit Depth Imagery

Due to measurement noise and image contrast changes from image to image, the same pixel location may have different pixel intensity values in different images. This difference can be enhanced in low bit depth images, due to the quantisation process, if the image intensity values fluctuate near a quantisation threshold. Then the image registration process is carried out to align similarity distorted images using the PC method.

To simulate a sequence of images taken by sensors, each of the two 8-bit depth images shown in Figure 5.1a and 5.1c are distorted by a random similarity transformation (i.e. a random value in translation, scale and rotation). To model the measurement noise, Gaussian noise ($\sigma = 2^4 = 16$ grey-levels, i.e. the 4 LSBs) is added to each distorted image. Also to model the image contrast changes from image to image, the quantisation thresholds for each of the distorted images are randomly shifted up or down by a small amount (by 16 grey-levels uniformly distributed and applied globally to every pixel), but the grey-levels of images after quantisation are always kept at 32, 96, 160, and 224 when reconstructing the images. Nine 2-bit depth images are constructed for each of the two example 8-bit depth images, corresponding to nine random similarity transformations, as shown in Figure 5.2a-5.2i and 5.3a-5.3i respectively.

Registration of Low Bit Depth Imagery

The similarity distorted 2-bit depth images shown in Figure 5.2 and 5.3 are each registered according to the respective 2-bit depth images shown in Figure 5.1b and 5.1d, which are acted as the reference images. The PC method is applied to estimate the parameters of the similarity transformation for each of the distorted images. Finally, each distorted image is transformed based on the estimated transformation mapping. The registration results are shown in Figure 5.4.

Fusing of Low Bit Depth Imagery

The two sets of nine aligned images are each fused together to obtain the corresponding reconstructed image, as shown in Figure 5.6. From the fusion results, it could be found that the reconstructed images not only contain almost the same detail compared to the

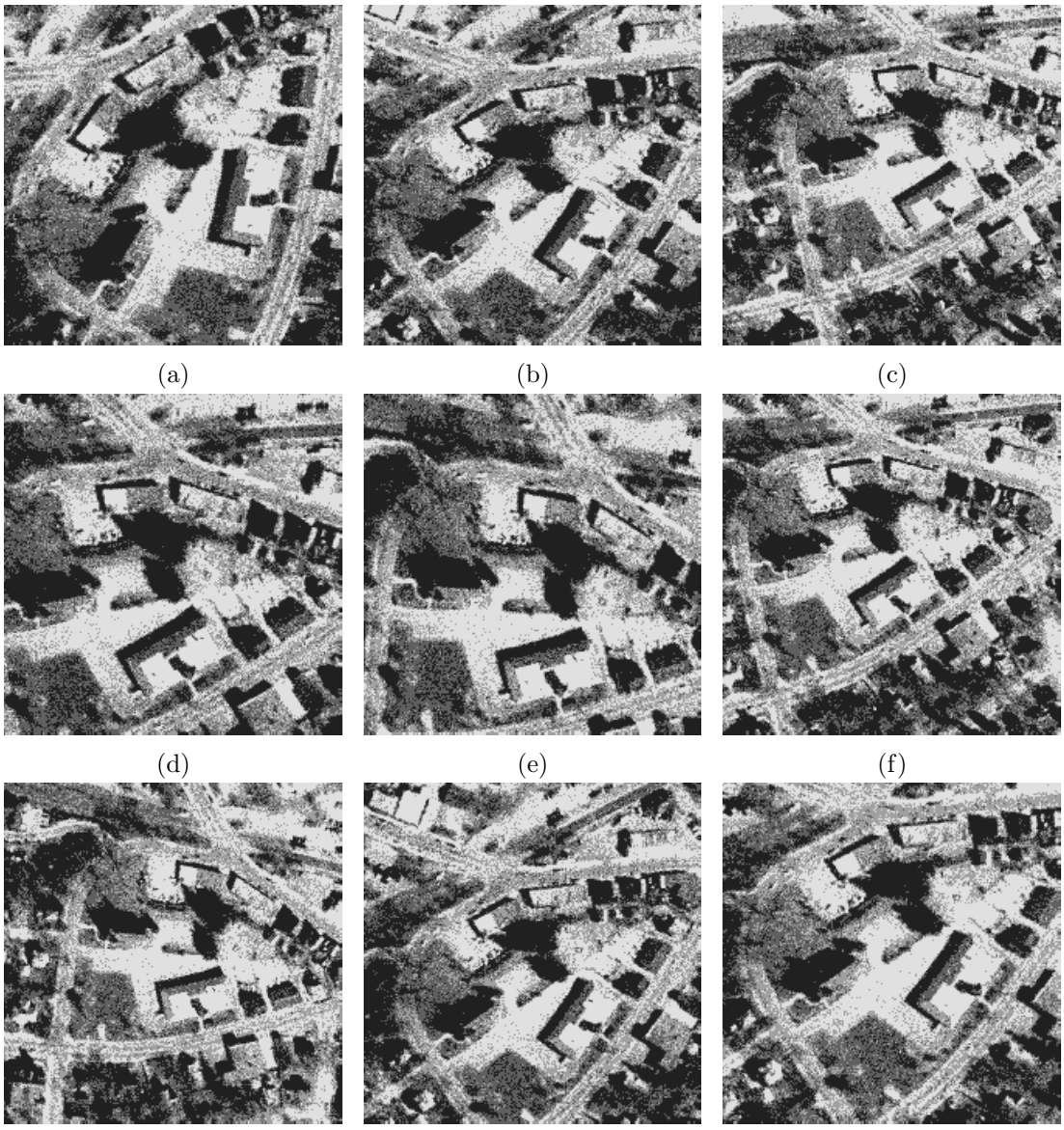


Figure 5.2: A sequence of nine similarity distorted 2-bit depth images of the 8-bit depth example image shown in Figure 5.1a (i.e. aerial image).

8-bit depth ones shown in Figure 5.1a and 5.1c, but also appear smoother (i.e. have less noise).

5.3.1 Reconstruction Quality Assessment

Two evaluate the reconstructed images, two measures of image quality are employed here. One is the Normalised Cross Correlation (NCC), which has been reviewed in section 3.3. The other is the Structural SIMilarity (SSIM) index [294,295], which treats image degradation to be perceived change in structural information. It is based on the fact that the pixel locations close by in the spatial domain have strong dependencies

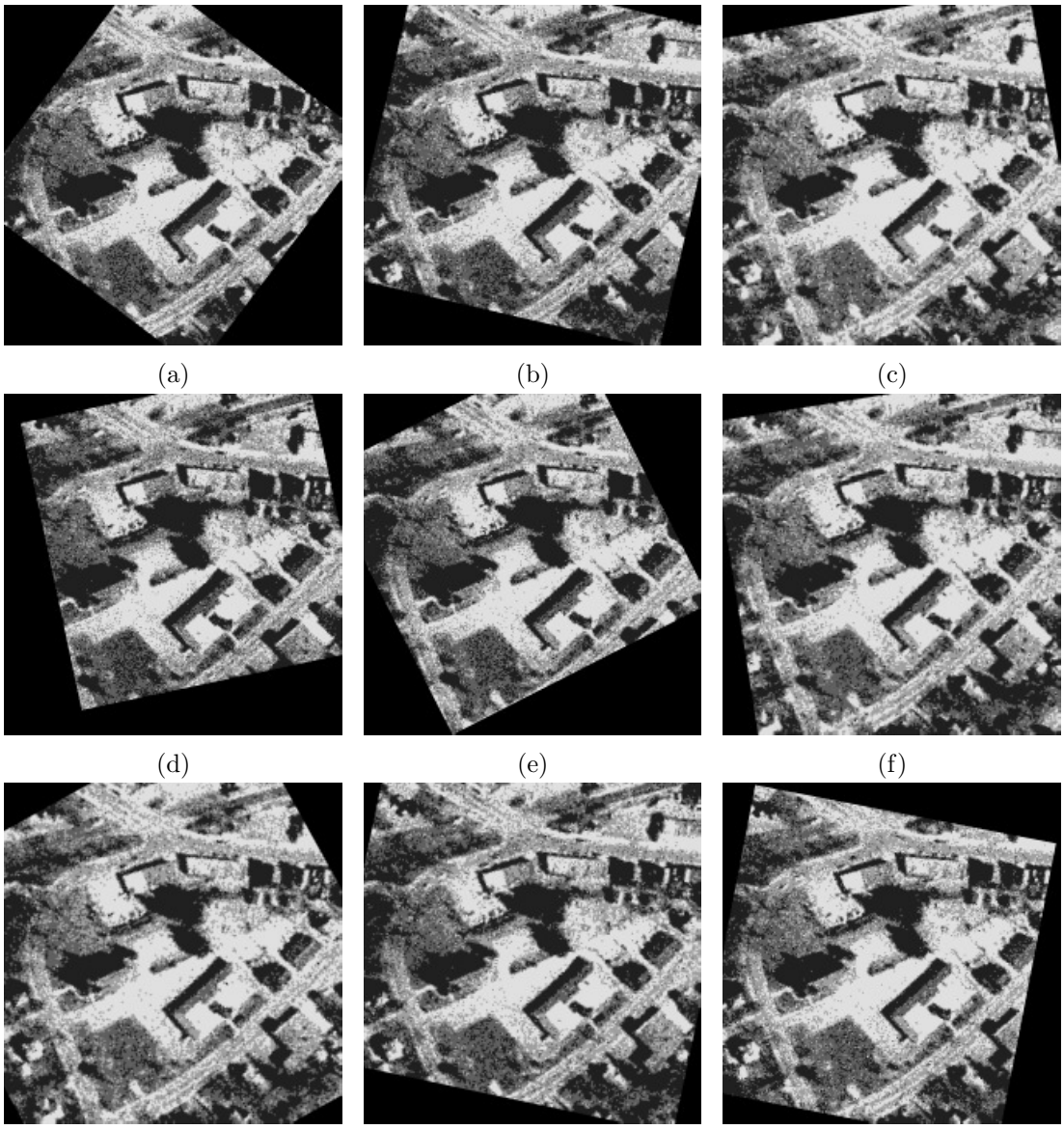
Figure 5.3: A sequence of nine similarity distorted 2-bit depth images of the 8-bit depth example image shown in Figure 5.1c (i.e. Lena).

between each other, which carry important information about the structure of an object. For two images I_a and I_b, the SSIM index is calculated as [295]:

$$\text{SSIM}(I_a, I_b) = l_u(I_a, I_b)^\alpha \times c_o(I_a, I_b)^\beta \times s_t(I_a, I_b)^\gamma \tag{5.7}$$

where $\alpha > 0$, $\beta > 0$ and $\gamma > 0$ control the relative significance of each of the three terms in the index, and the three terms (i.e. the luminance, contrast, and structural components) are defined respectively as [295]:

$$l_u(I_a, I_b) = \frac{2\mu_{I_a}\mu_{I_b} + c_1^c}{\mu_{I_a}^2 + \mu_{I_b}^2 + c_1^c} \tag{5.8}$$



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.4: Registration results of the images shown in Figure 5.2.

$$c_o(I_a, I_b) = \frac{2\sigma_{I_a}\sigma_{I_b} + c_2^c}{\sigma_{I_a}^2 + \sigma_{I_b}^2 + c_2^c} \tag{5.9}$$

$$s_t(I_a, I_b) = \frac{\sigma_{I_a I_b} + c_3^c}{\sigma_{I_a}\sigma_{I_b} + c_3^c} \tag{5.10}$$

where μ_{I_a} and μ_{I_b} are the means of the respective I_a and I_b images, while σ_{I_a} and σ_{I_b} are the respective standard deviations, and $\sigma_{I_a I_b}$ is the covariance between the two images. To prevent the situation where the denominators are weak (i.e. close to zero), the three constants c_1^c, c_2^c and c_1^c are introduced [294], and they are defined respectively as [295]:

$$c_1^c = (k_1^c l^c)^2$$

Figure 5.5: Registration results of the images shown in Figure 5.3.

$$c_2^c = (k_2^c l^c)^2$$
$$c_3^c = \frac{c_2^c}{2} \tag{5.11}$$

where l^c is the dynamic range of the pixel intensity values (e.g. 255 for a 8-bit depth image), while $k_1^c = 0.01$ and $k_2^c = 0.03$ [296].

To assess the quality of the reconstructed images, and in addition, to show the reason why 2-bit depth imagery is chosen to be the low bit depth imagery, the measures of NCC and SSIM were carried out between the original and reconstructed images. To generalise the measurement results, for each variable bit (i.e. from 1-bit to 7-bit), the above reconstruction process has been repeated by 100 times for different aerial images. The results are displayed in Figure 5.7. From Figure 5.7, it can be seen that all the

(a) original aerial image (b) reconstructed aerial image

(c) original Lena (d) reconstructed Lena

Figure 5.6: Original 8-bit depth images, and reconstructed results by fusing the corresponding aligned images shown in Figure 5.4 and 5.5.

reconstructions achieve almost the same performance, except the one from the 1-bit depth imagery. As a result, choosing 2-bit depth imagery as the low bit depth imagery can save more in the computational load and transmission bandwidth, while the final reconstruction result is comparable to the ones based on higher bit depth imagery.

5.3.2 Further Reconstruction Quality Assessment

To evaluate the reliability of the proposed method, its performance is further investigated against different levels of noise and blurring. Here, to simplify the evaluation process, Gaussian noise is considered as the noise, while Gaussian blur (i.e. the result of blurring an image by a Gaussian kernel), which is used in (3.57) to generate scale-space

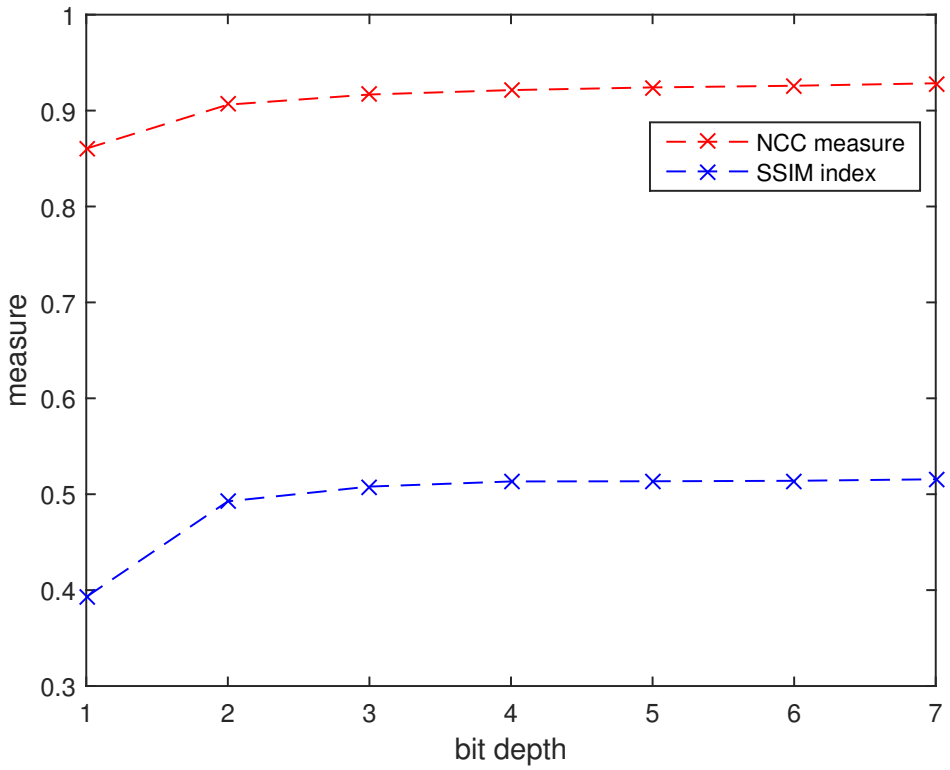
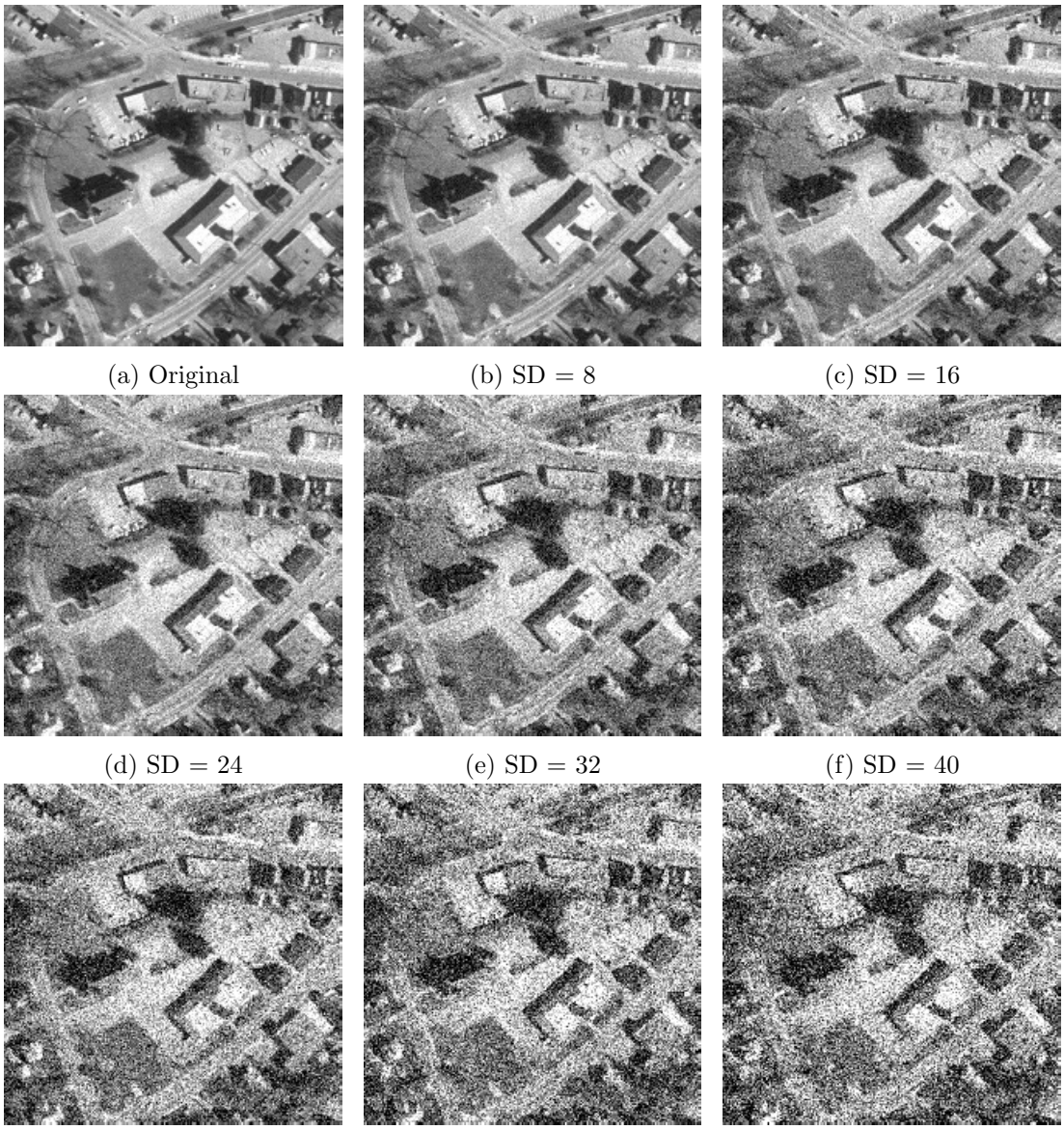


Figure 5.7: The NCC and SSIM measures between the original and reconstructed images: the reconstruction process was repeated 100 times, for different bit depth images (i.e. from 1-bit to 7-bit).

representation, is considered as the blurring.

Eight different levels of noise are considered here, and it is achieved by varying the Standard Deviation (SD) of Gaussian noise from 8 to 64, with the step value being set to be 8. Whilst four different levels of blurring are considered here, and it is achieved by setting the scale σ in (3.58) to be $1.6/\sqrt{2}$, 1.6, $1.6 \times \sqrt{2}$ and 1.6×2, respectively. Additionally, the case of no blurring is also considered. To visually show the effects of different levels of noise and blurring, examples of an example image under the eight different levels of Gaussian noise and five different levels of blurring (i.e. the four different levels of Gaussian blur, as well as the case of no blurring) are shown in Figure 5.8, Figure 5.9, Figure 5.10, Figure 5.11 and Figure 5.12, respectively. From Figure 5.8, Figure 5.9, Figure 5.10, Figure 5.11 and Figure 5.12, it can be seen that with deeper levels of noise and/or blurring, more image details are lost. Additionally, for image under higher levels of Gaussian blur, the different levels of Gaussian noise have smaller effects on the structural information in images.

To fully investigate the performance of the proposed method, the performances over changes of bit depth (i.e. the different bit depth images from 1-bit to 7-bit) were evaluated under the above different levels of noise and blurring. The evaluation results



(a) Original (b) SD = 8 (c) SD = 16

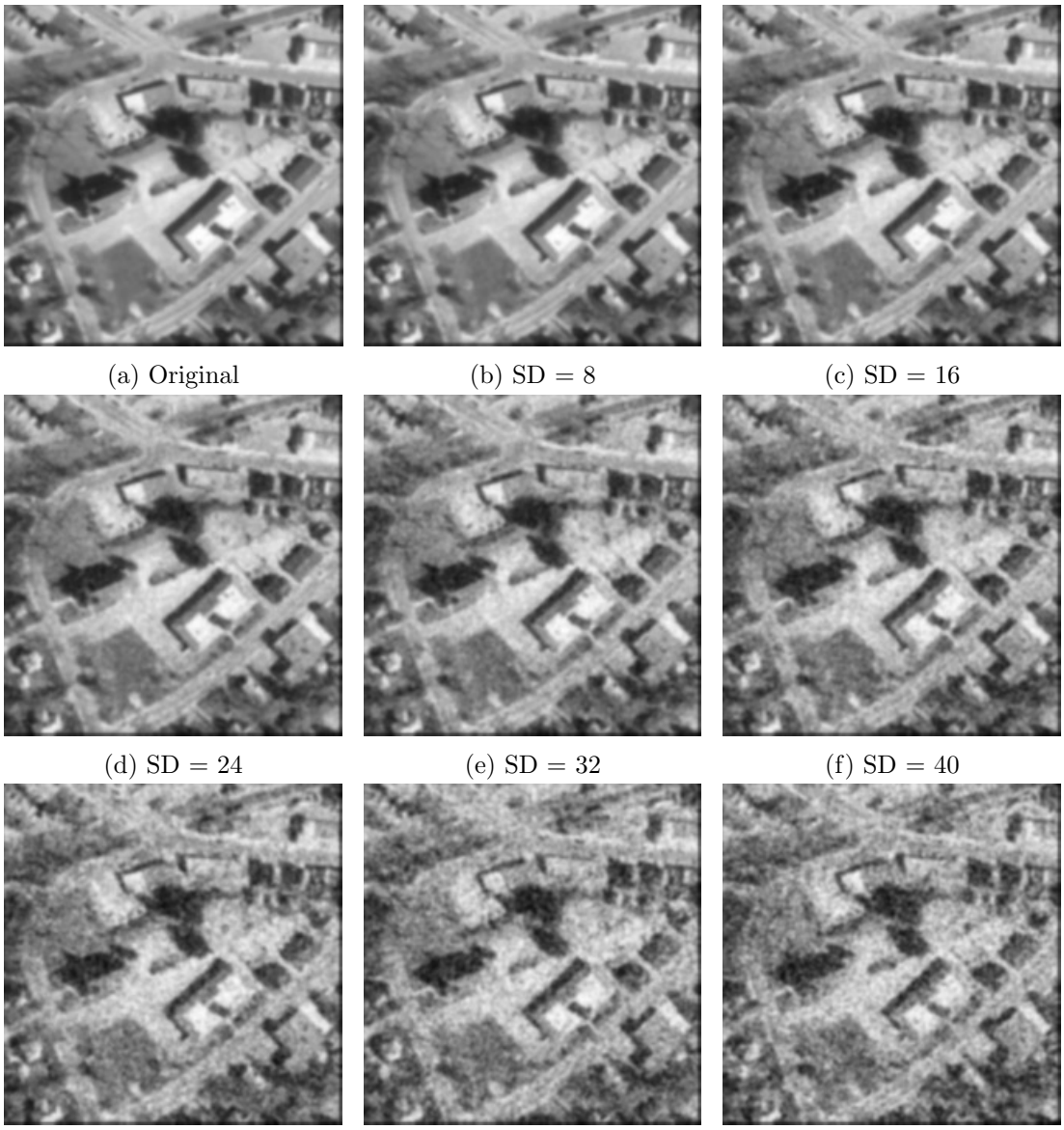
(d) SD = 24 (e) SD = 32 (f) SD = 40

(g) SD = 48 (h) SD = 56 (i) SD = 64

Figure 5.8: Examples of an example under different levels of Gaussian noise and without any blurring: the SD of Gaussian noise is varied from 8 to 64, with a constant step value of 8.

for 1-bit to 7-bit depth images are shown in Figure 5.13, Figure 5.14, Figure 5.15, Figure 5.16, Figure 5.17, Figure 5.18 and Figure 5.19, respectively.

From Figure 5.13a, Figure 5.14a, Figure 5.15a, Figure 5.16a, Figure 5.17a, Figure 5.18a and Figure 5.19a, it can be seen that for the different bit depth images from 1-bit to 7-bit, the NCC measures corresponding to the case of no blurring possess the best performances. Additionally, the NCC measures corresponding to the case of no blurring and all the four blurring case first increase to reach peak values and then drop as the SD of Gaussian noise increase. For the case of 1-bit depth images, the best



(a) Original (b) SD = 8 (c) SD = 16

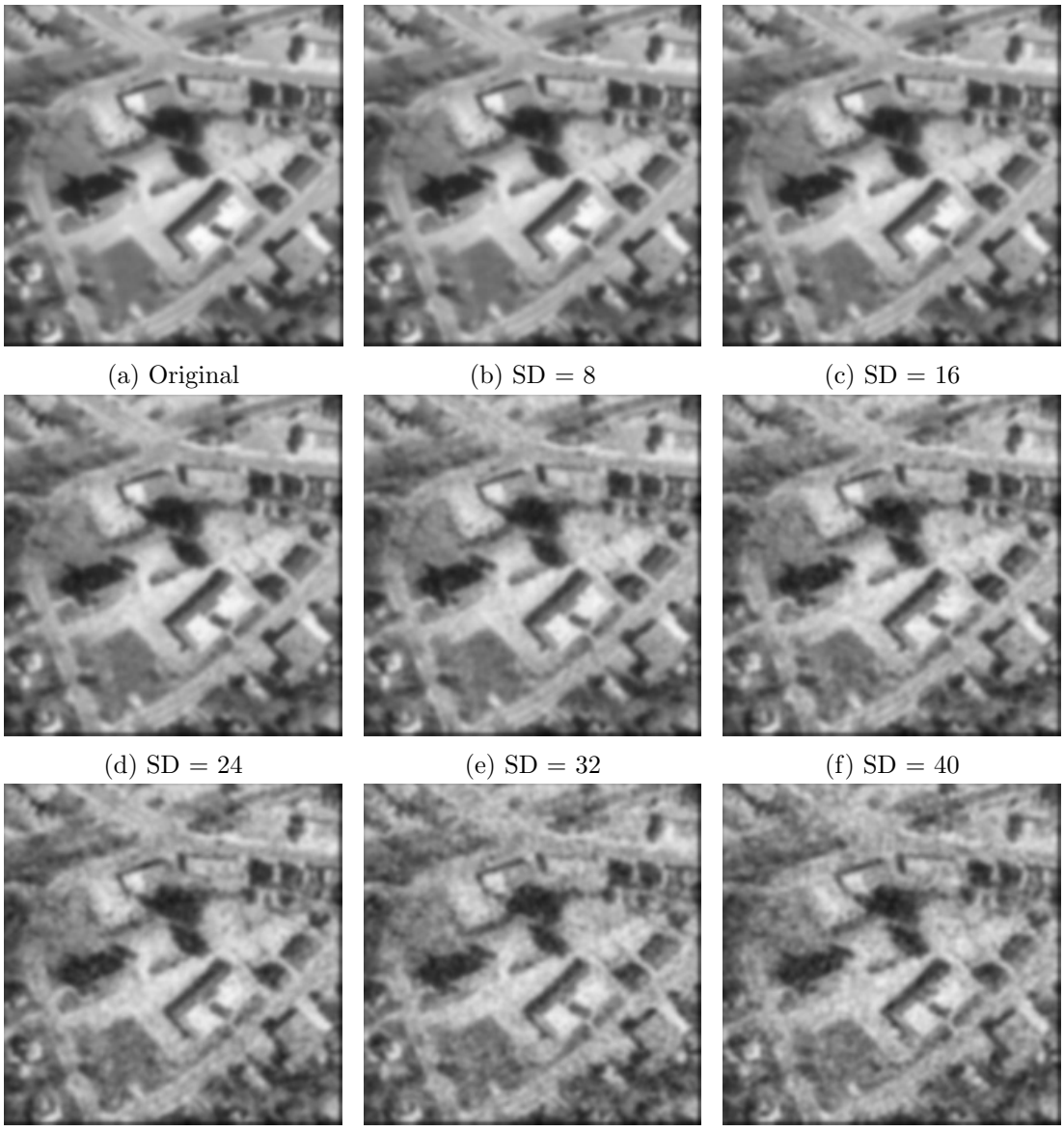
(d) SD = 24 (e) SD = 32 (f) SD = 40

(g) SD = 48 (h) SD = 56 (i) SD = 64

Figure 5.9: Examples of an example image under different levels of Gaussian noise and a fixed Gaussian blur (i.e. the scale $\sigma = 1.6/\sqrt{2}$): the SD of Gaussian noise is varied from 8 to 64, with a constant step value of 8.

NCC measure is reached when the SD of Gaussian noise is 48. While for the cases of 2-bit, 4-bit and 6-bit depth images, the best NCC measures are reached when the SD of Gaussian noise is 24. Then for the cases of 3-bit and 5-bit depth images, the best NCC measures are reached when the SD of Gaussian noise is 16. Finally, for the case of 7-bit depth images, the best NCC measure is reached when the SD of Gaussian noise is 32.

From Figure 5.13b, Figure 5.14b, Figure 5.15b, Figure 5.16b, Figure 5.17b, Figure 5.18b and Figure 5.19b, it can be seen that for the different bit depth images from



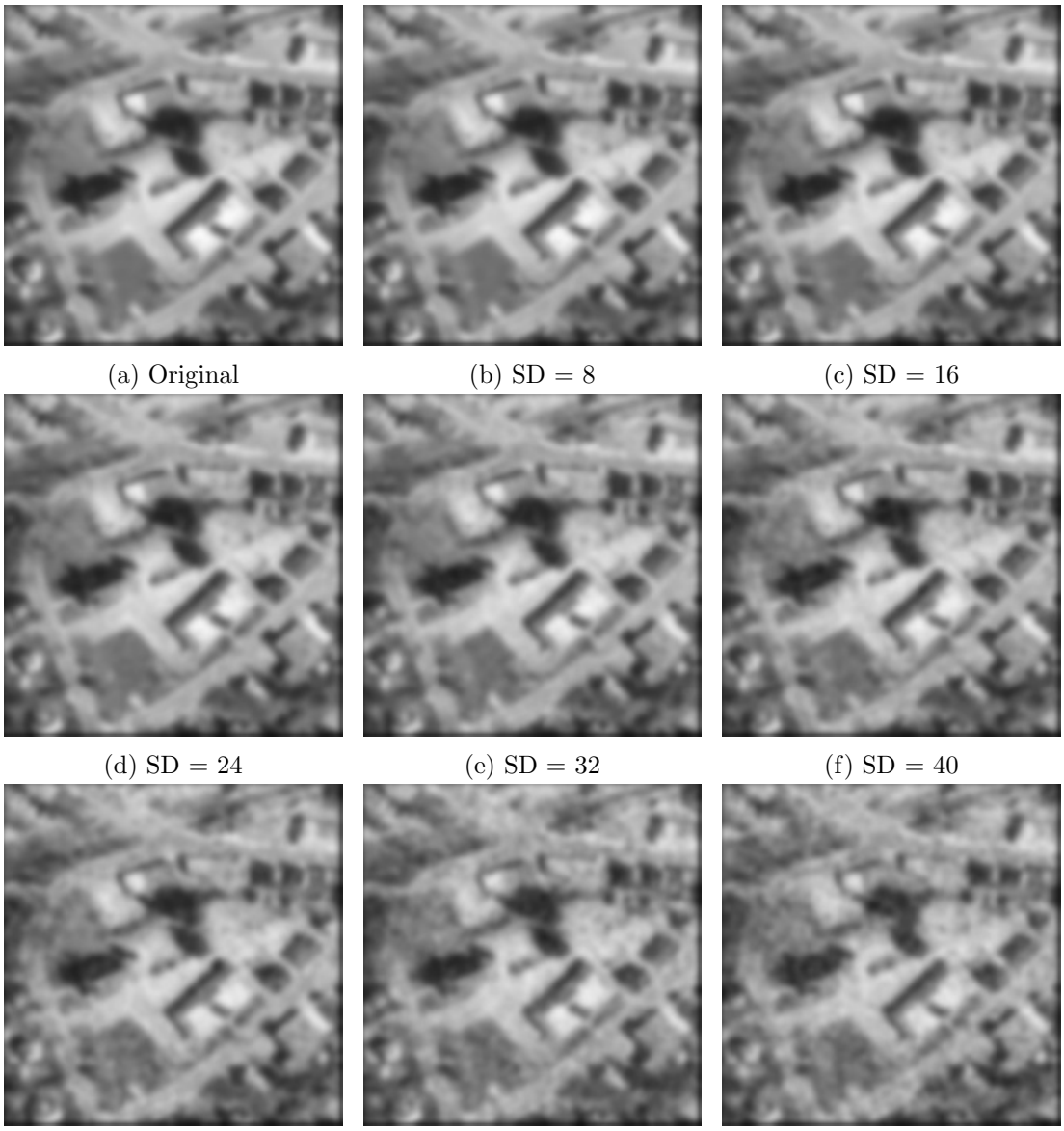
(a) Original (b) SD = 8 (c) SD = 16

(d) SD = 24 (e) SD = 32 (f) SD = 40

(g) SD = 48 (h) SD = 56 (i) SD = 64

Figure 5.10: Examples of an example image under different levels of Gaussian noise and a fixed Gaussian blur (i.e. the scale $\sigma = 1.6$): the SD of Gaussian noise is varied from 8 to 64, with a constant step value of 8.

1-bit to 7-bit, the SSIM indices corresponding to all the four blurred versions of images are almost flat as the SD of Gaussian noise increases. It is due to the fact that the structural information in images is vanished because of the blurring process, regardless of the present of different levels of noise. Additionally, for all the four blurred versions of images, the SSIM indices corresponding to higher bit depth images are slightly better compared to those corresponding to lower bit depth images. It is because that higher bit depth images corresponding to a smaller quantisation range, which results of preserving more structural information in images. Whilst for the case of no blurring, the SSIM



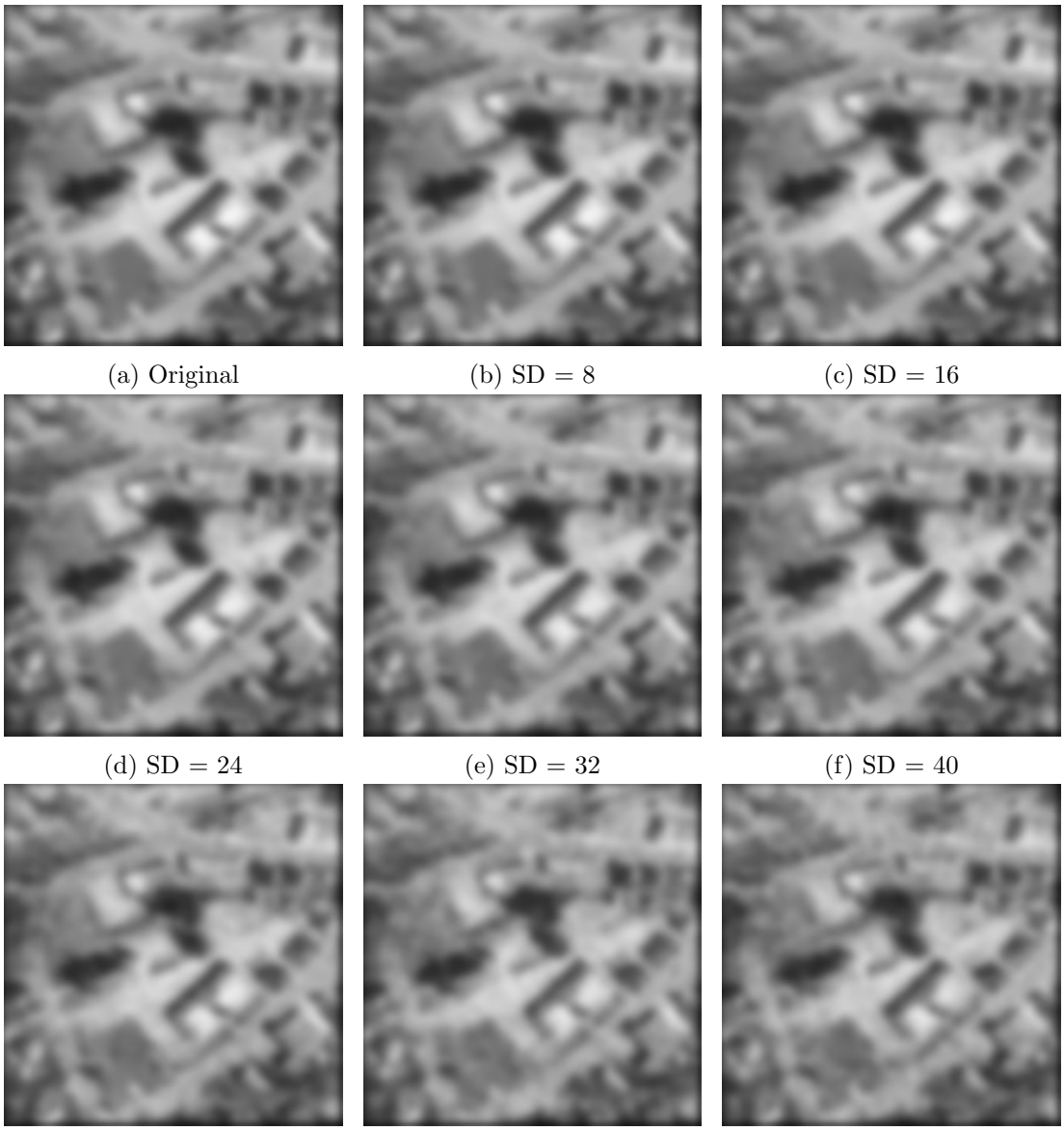
(a) Original (b) SD = 8 (c) SD = 16

(d) SD = 24 (e) SD = 32 (f) SD = 40

(g) SD = 48 (h) SD = 56 (i) SD = 64

Figure 5.11: Examples of an example image under different levels of Gaussian noise and a fixed Gaussian blur (i.e. the scale $\sigma = 1.6 \times \sqrt{2}$): the SD of Gaussian noise is varied from 8 to 64, with a constant step value of 8.

indices corresponding to the different bit depth images from 1-bit to 7-bit possess the best performances, and they first increase to reach peak values and then drop as the SD of Gaussian noise increases. For the different bit depth images from 1-bit to 7-bit, the peak values of SSIM indices are reached when the SD of Gaussian noise is 16 except for the case of 1-bit depth images, where the peak value of SSIM index is reached when the SD of Gaussian noise is 24. Additionally, when the SD of Gaussian noise increases to higher levels, the SSIM indices corresponding to the different bit depth images from 1-bit to 7-bit drop so fast that they are even worse compared to those corresponding



(a) Original (b) SD = 8 (c) SD = 16

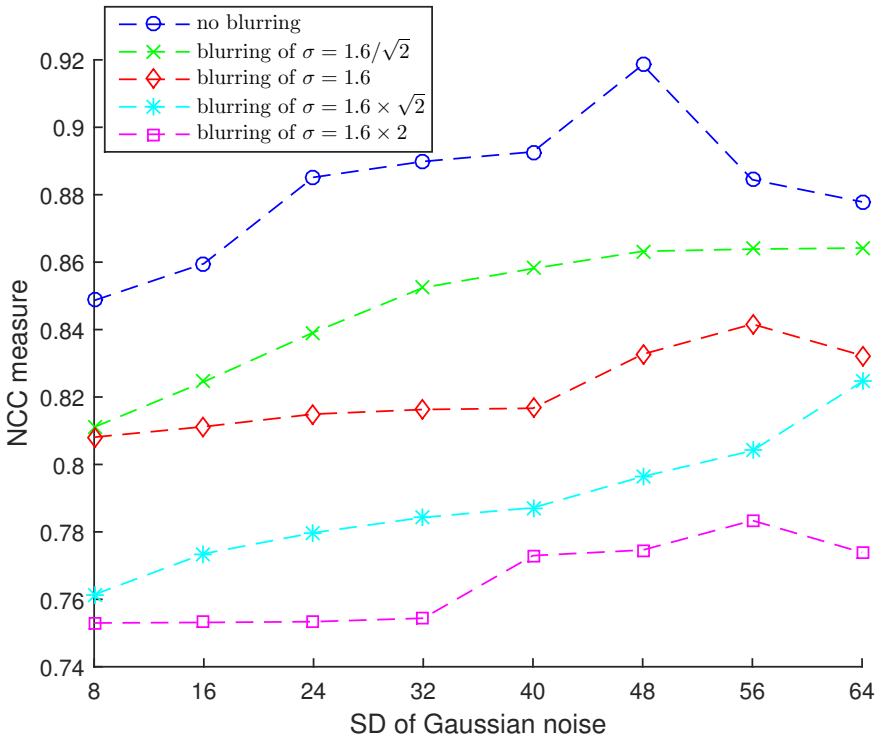
(d) SD = 24 (e) SD = 32 (f) SD = 40

(g) SD = 48 (h) SD = 56 (i) SD = 64

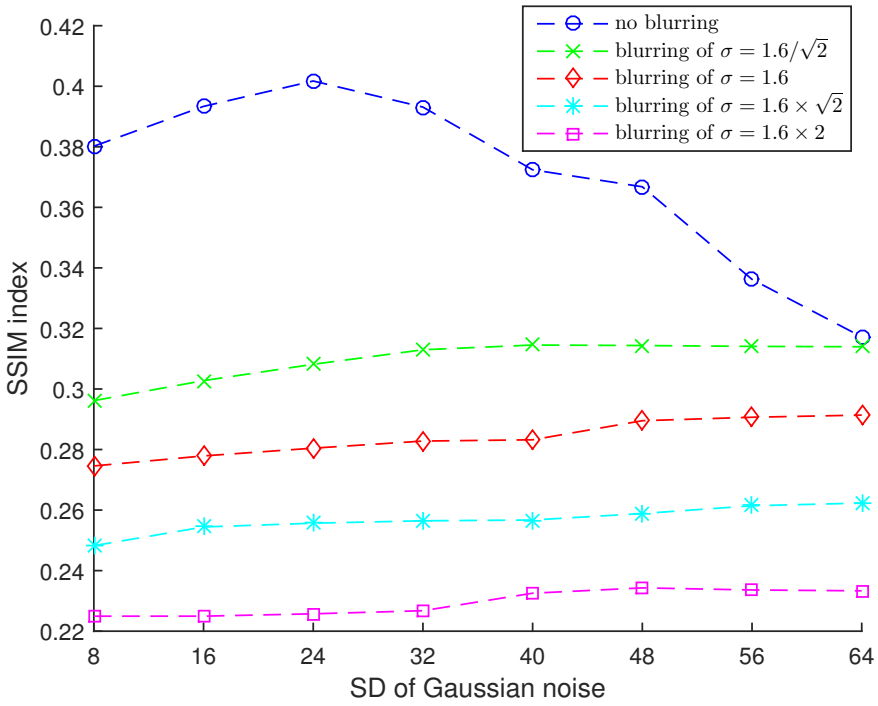
Figure 5.12: Examples of an example image under different levels of Gaussian noise and a fixed Gaussian blur (i.e. the scale $\sigma = 1.6 \times 2$): the SD of Gaussian noise is varied from 8 to 64, with a constant step value of 8.

to all the four blurring cases.

To sum up, the proposed method is able to work under the above different levels of Gaussian noise and Gaussian blur, which demonstrates the reliability of the proposed method. Although the present of Gaussian noise and/or Gaussian blur tends to remove image details (e.g. the structural information in images), for the different bit depth images from 1-bit to 7-bit and the different levels of blurring, there is always a certain level of noise that can make the proposed method to achieve the best performance. It results of that the proposed method can work well, as long as there exists a reasonable

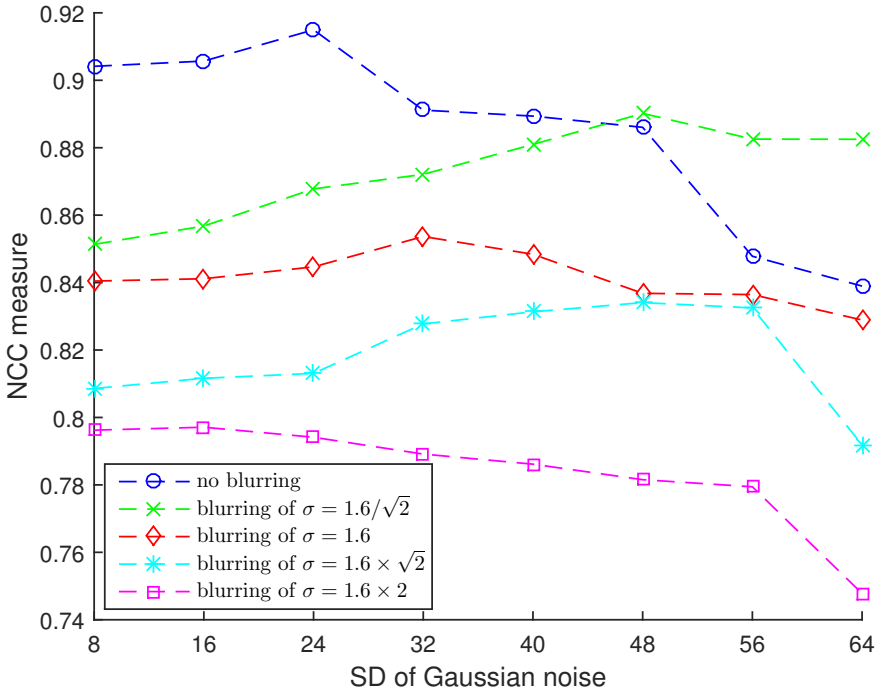


(a)

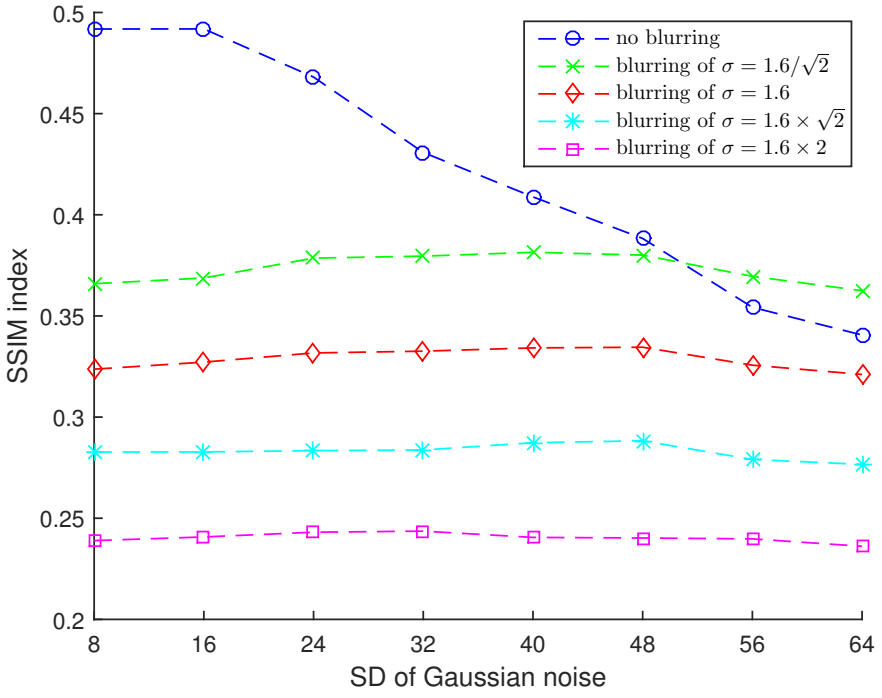


(b)

Figure 5.13: The NCC and SSIM measures between the original and reconstructed images: the reconstruction process was repeated 100 times, for different 1-bit depth images.

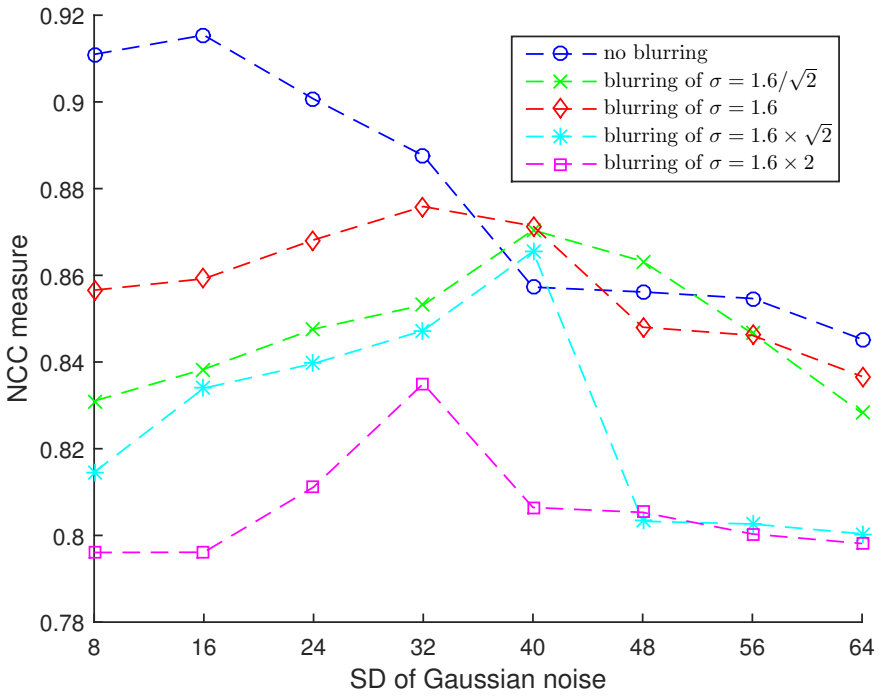


(a)

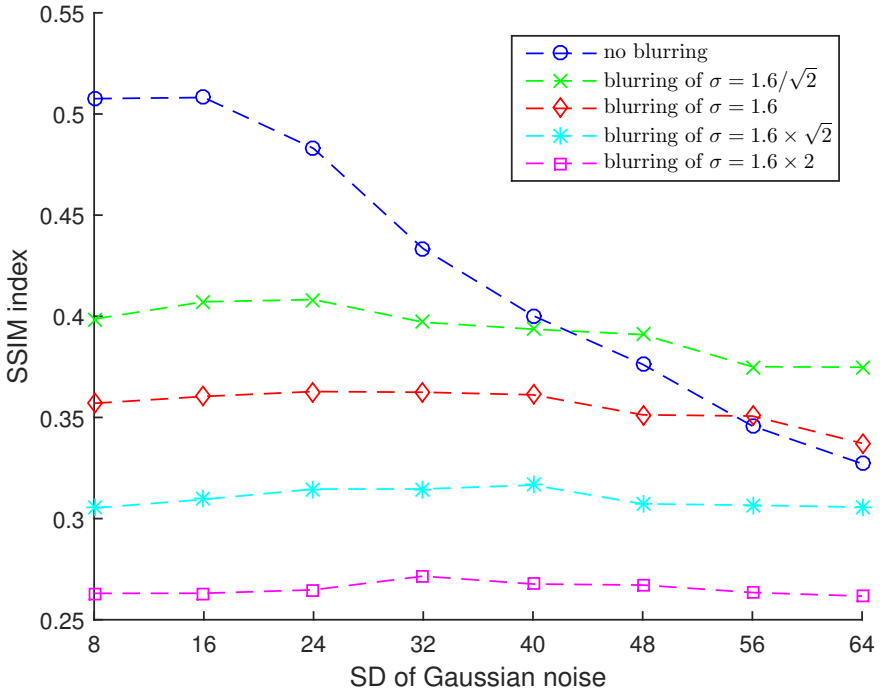


(b)

Figure 5.14: The NCC and SSIM measures between the original and reconstructed images: the reconstruction process was repeated 100 times, for different 2-bit depth images.

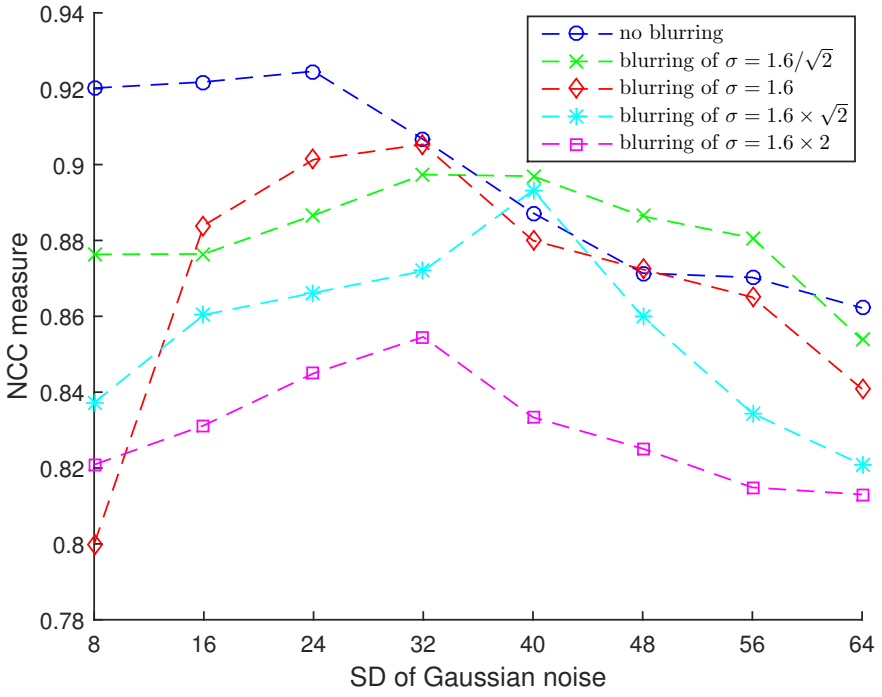


(a)

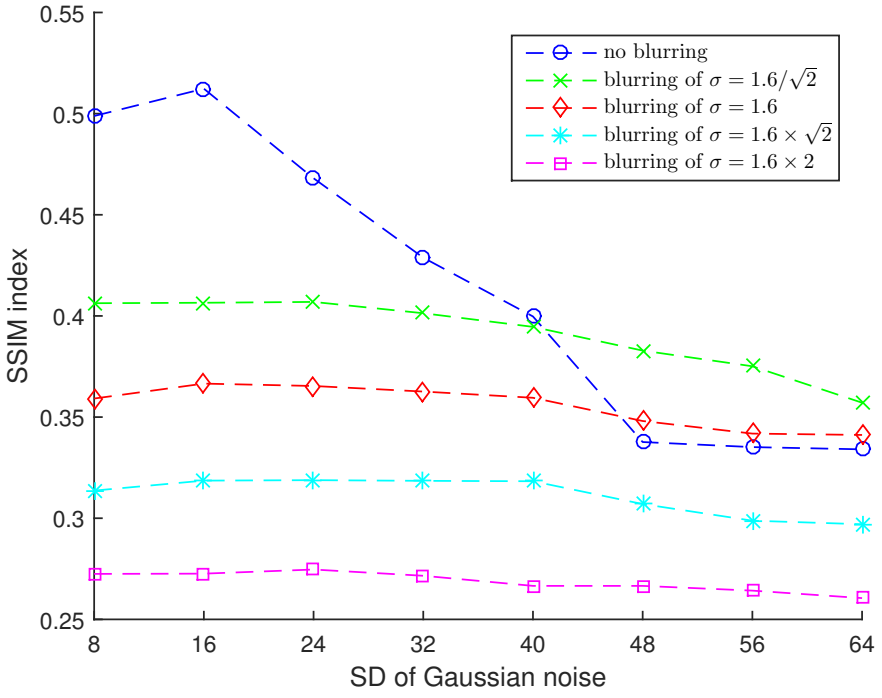


(b)

Figure 5.15: The NCC and SSIM measures between the original and reconstructed images: the reconstruction process was repeated 100 times, for different 3-bit depth images.

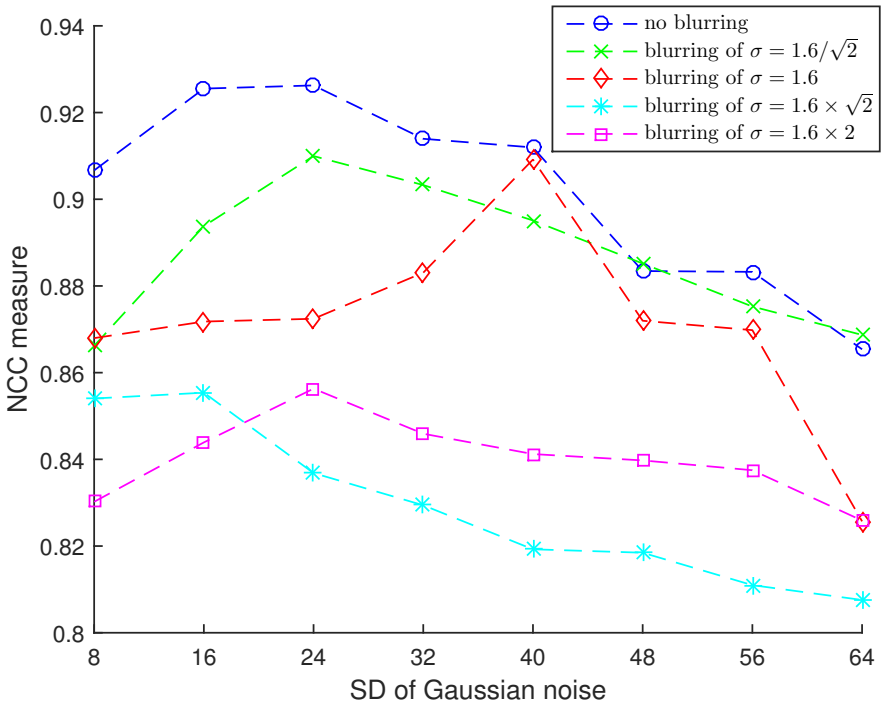


(a)

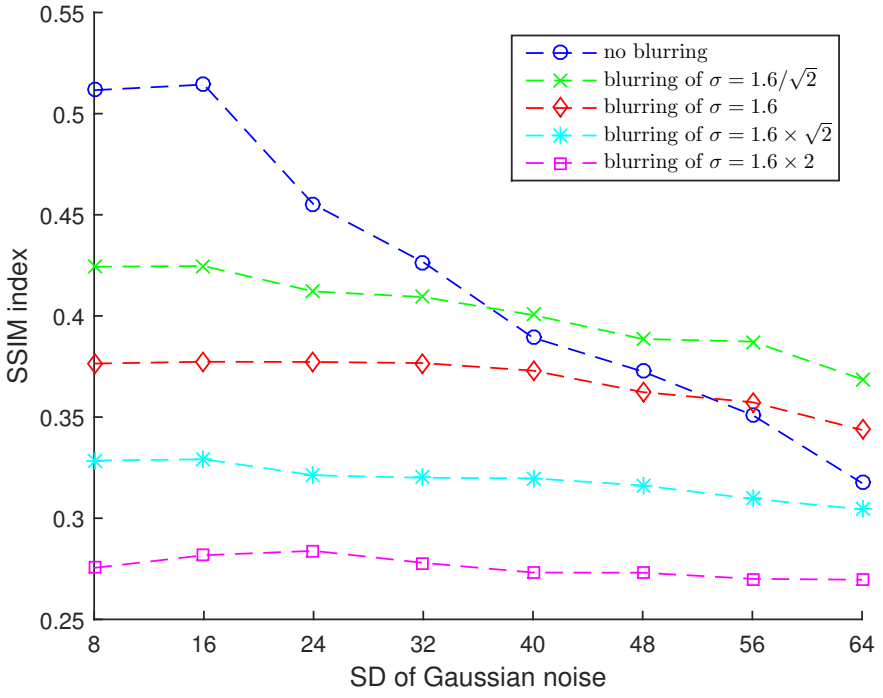


(b)

Figure 5.16: The NCC and SSIM measures between the original and reconstructed images: the reconstruction process was repeated 100 times, for different 4-bit depth images.

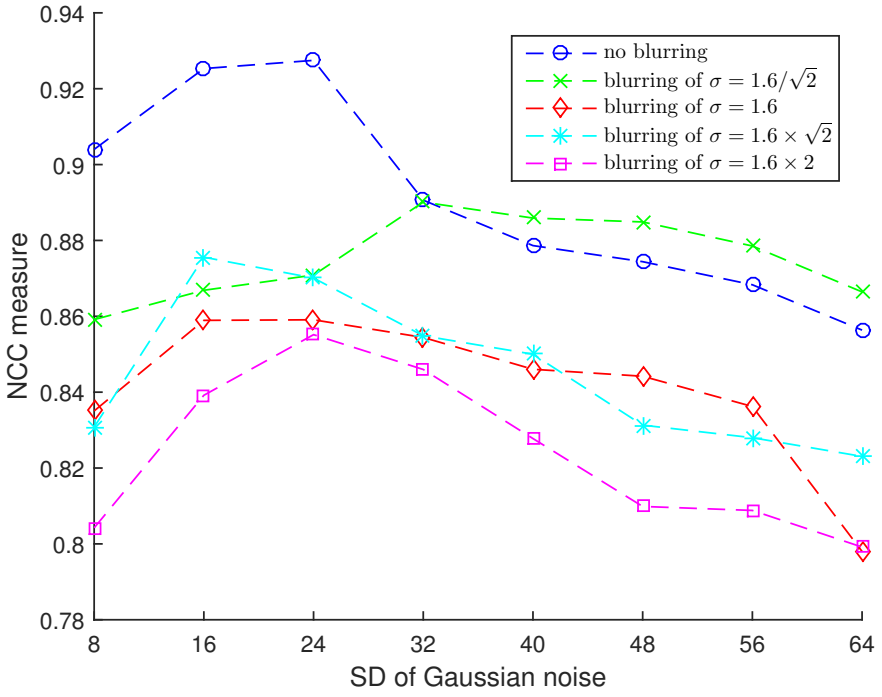


(a)

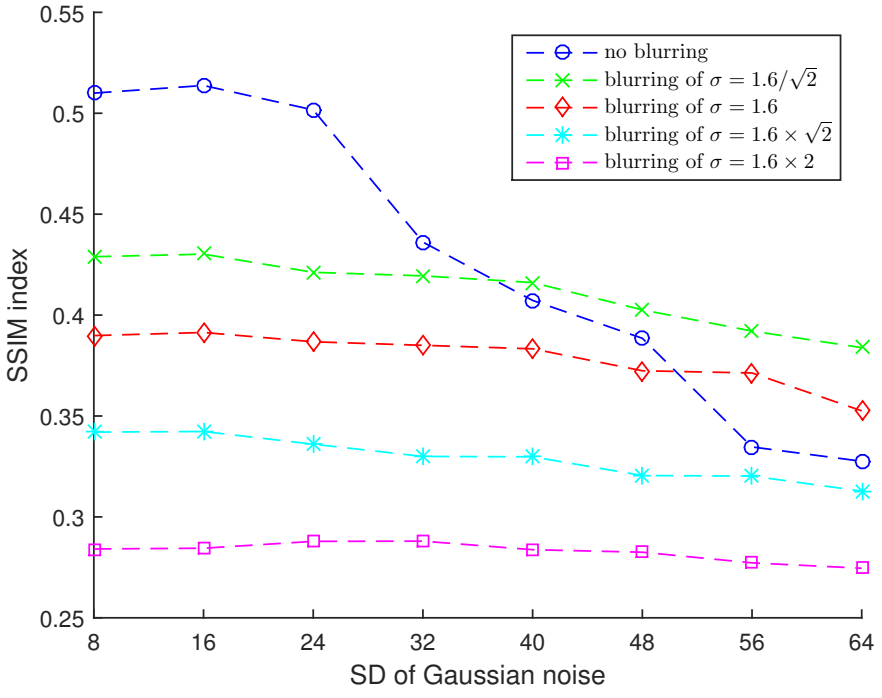


(b)

Figure 5.17: The NCC and SSIM measures between the original and reconstructed images: the reconstruction process was repeated 100 times, for different 5-bit depth images.

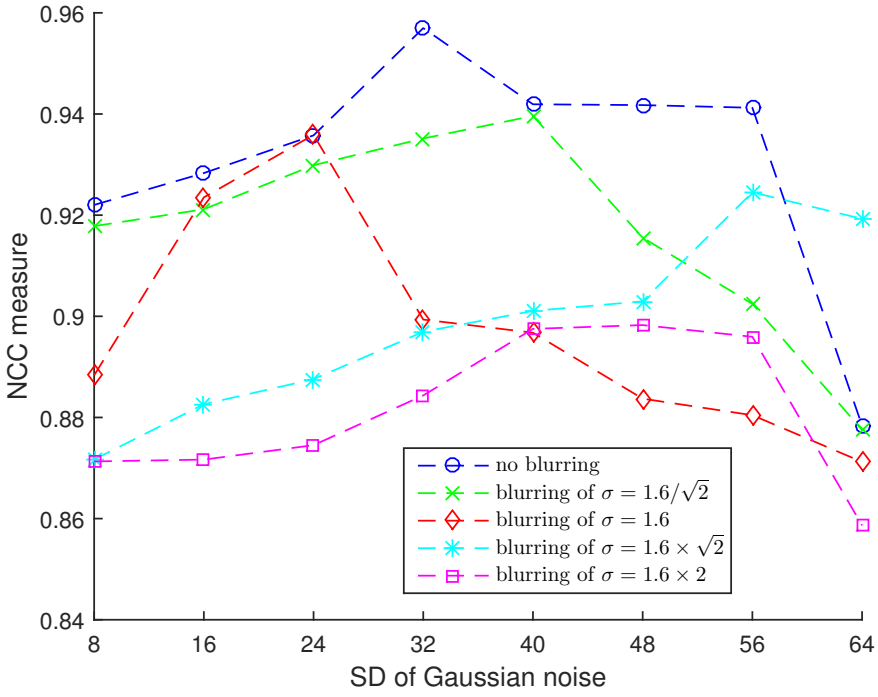


(a)

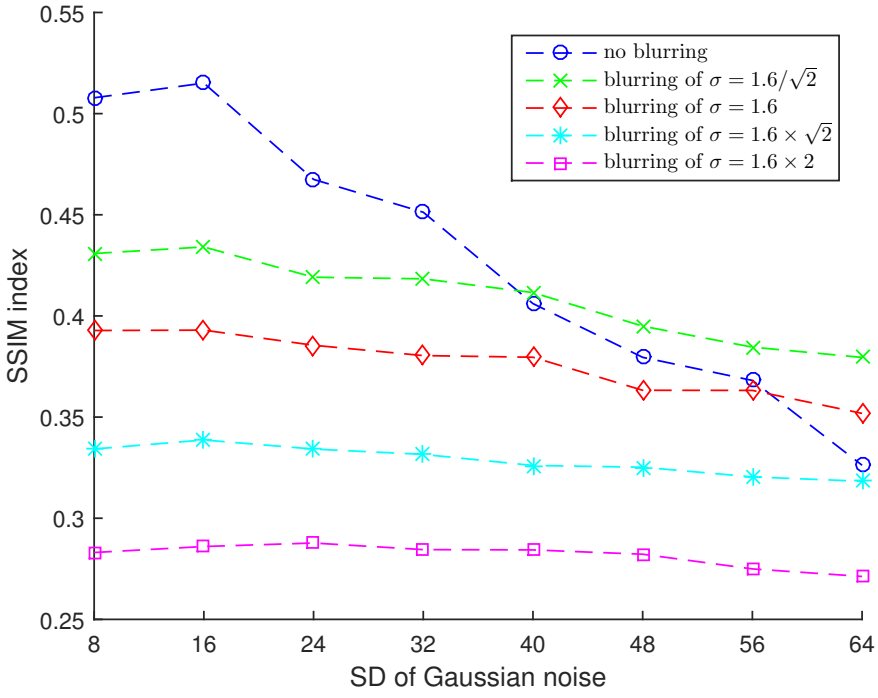


(b)

Figure 5.18: The NCC and SSIM measures between the original and reconstructed images: the reconstruction process was repeated 100 times, for different 6-bit depth images.



(a)



(b)

Figure 5.19: The NCC and SSIM measures between the original and reconstructed images: the reconstruction process was repeated 100 times, for different 7-bit depth images.

level of noise. Meanwhile, the process of Gaussian blur can somehow preserve some certain structural information in images, regardless of the present of different levels of Gaussian noise. It results of that the process of Gaussian blur may make a positive contribution to the proposed method, especially when there is a high level of Gaussian noise present in images.

Chapter 6

3D Scene Reconstruction from 2D images

6.1 Motivation

In the past decade, there has been a technological revolution in 3D scene geometry, due to the significant progress in view synthesis methods by research community and commercial products [233]. 3D scene geometry has a broad range of application areas, e.g. Digital Elevation Models (DEMs), which is a 3D model of a terrain's surface constructed based on terrain elevation data [23]. There are various ways to capture 3D scene geometry, such as laser scanning [297, 298], and photogrammetry [299]. However, to facilitate 3D scene reconstruction, traditional photogrammetric techniques need the 3D location and pose of each camera to be known [23]. While the technique called Structure from Motion (SfM) can simultaneously and automatically solve the camera 3D location and pose as well as 3D scene geometry, based on a number of overlapping and offset 2D images. In general, a sparse point cloud generated by SfM technique, and further its density is improved by a dense reconstruction, e.g. Clustering Views for Multi-view Stereo (CMVS) [250, 251], is sufficient for most 3D reconstruction tasks. Both sparse and dense point cloud reconstructions have been reviewed in section 3.7.1. However, in some situations where there are a limited number of images available and/or most of the images are taken from a similar angles, or the applications focus on the reconstruction details, this approach will not provide a good solution. It is because when the number of images and/or views are limited, there will be a limited number of 3D scene geometry available for capturing, which results of that the density of the point cloud generated by SfM is also limited.

Two-dimensional images are captured by a mapping from the 3D scene to 2D image planes, it is possible that parts of 2D images can be mapped to the corresponding areas in the 3D scene. A common way to approach this is to form a large number of triangles based on the sparse point cloud generated by SfM, and then mapping 2D image triangles back to the 3D scene one by one. Although this approach can achieve

good reconstruction performance, it is quite computationally expensive. Here, a new approach has been proposed to achieve a good balance between the computational load and the reconstruction performance. For a flat or almost flat region, several large triangles each of which covers a number of small triangles will be used instead of the small triangles to describe the region. The large triangles are formed based on the boundary points of this region.

6.2 3D Reconstruction Process

The 3D reconstruction process developed here is based on a sparse point cloud generated by employing the Structure from Motion (SfM) technique, based on a number of 2D images of the same scene. The SfM technique has been reviewed in section 3.7.1. The reconstruction process consists of the following 4 stages: Determining Valid Planes, Defining Flat Regions, Forming Large Triangles, and Mapping 2D Triangles.

Determining Valid Planes

A commonly used approach to reconstruct a 3D scene is by capturing triangles in 2D images, and then mapping them back to the corresponding positions in the 3D scene. This is based on the fact that arbitrary three non-collinear 3D points (i.e. points not on a single line) can uniquely determine a plane. Then the mapping process is simplified to be a transformation between two planes (i.e. from an image plane to this determined plane). However, in real-time applications, not every combination of arbitrary three 3D points has its correspondence in the available 2D images. In other words, sometimes, there are some 3D points that have not been captured by the same image. Additionally, some 2D image triangles are meaningless to be mapped back, e.g. those that only have a small number of pixels. To prevent these situations, two constraints have been imposed when choosing a combination of three 3D points. Firstly, the three 3D points must be captured simultaneously in at least one 2D image. Secondly, for the corresponding triangle(s) in 2D images, its minimum height should greater than 1 pixel, and its minimum angle should greater than 10 degrees, so that the situation where there exists just few pixels can be prevented.

Defining Flat Regions

To reduce the total computational load, for flat and almost flat regions in the 3D scene, instead of mapping triangles one by one, a large triangle that consists of several triangles is determined. For a valid plane that is determined by three 3D points, the almost flat region can be determined by the 3D points lying on this plane and those lying close to this plane. The flatness of the region can be measured by calculating the variance of the distances of the 3D points in this region to this plane. For three 3D

points $P_a(x_{1_a}, x_{2_a}, x_{3_a})$, $P_b(x_{1_b}, x_{2_b}, x_{3_b})$ and $P_c(x_{1_c}, x_{2_c}, x_{3_c})$ that determines a valid plane $a_pX_1 + b_pX_2 + c_pX_3 + d_p = 0$, the parameters a_p, b_p, c_p and d_p can be calculated as:

$$\begin{aligned} a_p &= (x_{2_b} - x_{2_a})(x_{3_c} - x_{3_a}) - (x_{2_c} - x_{2_a})(x_{3_b} - x_{3_a}) \\ b_p &= (x_{3_b} - x_{3_a})(x_{1_c} - x_{1_a}) - (x_{3_c} - x_{3_a})(x_{1_b} - x_{1_a}) \\ c_p &= (x_{1_b} - x_{1_a})(x_{2_c} - x_{2_a}) - (x_{1_c} - x_{1_a})(x_{2_b} - x_{2_a}) \\ d_p &= -(a_p x_{1_a} + b_p x_{2_a} + c_p x_{3_a}) \end{aligned} \tag{6.1}$$

Then, for a point $P_d(x_{1_d}, x_{2_d}, x_{3_d})$, its distance to this plane d_{d2p} can be calculated as:

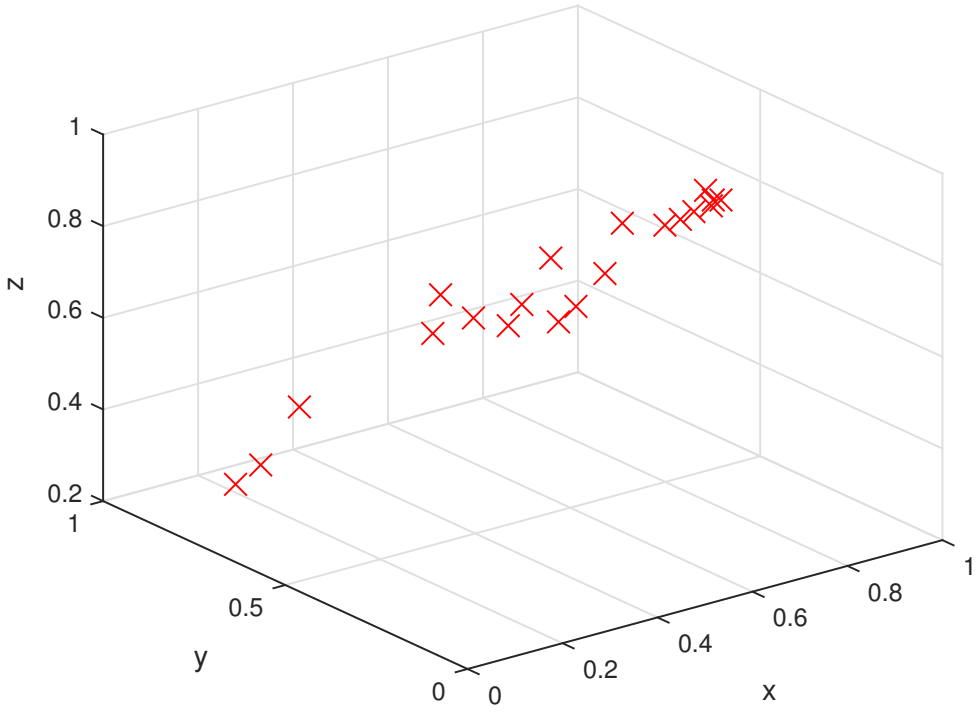
$$d_{d2p} = \frac{\|a_p x_{1_d} + b_p x_{2_d} + c_p x_{2_d} + d_p\|_1}{\sqrt{a_p^2 + b_p^2 + c_p^2}} \tag{6.2}$$

Forming Large Triangles

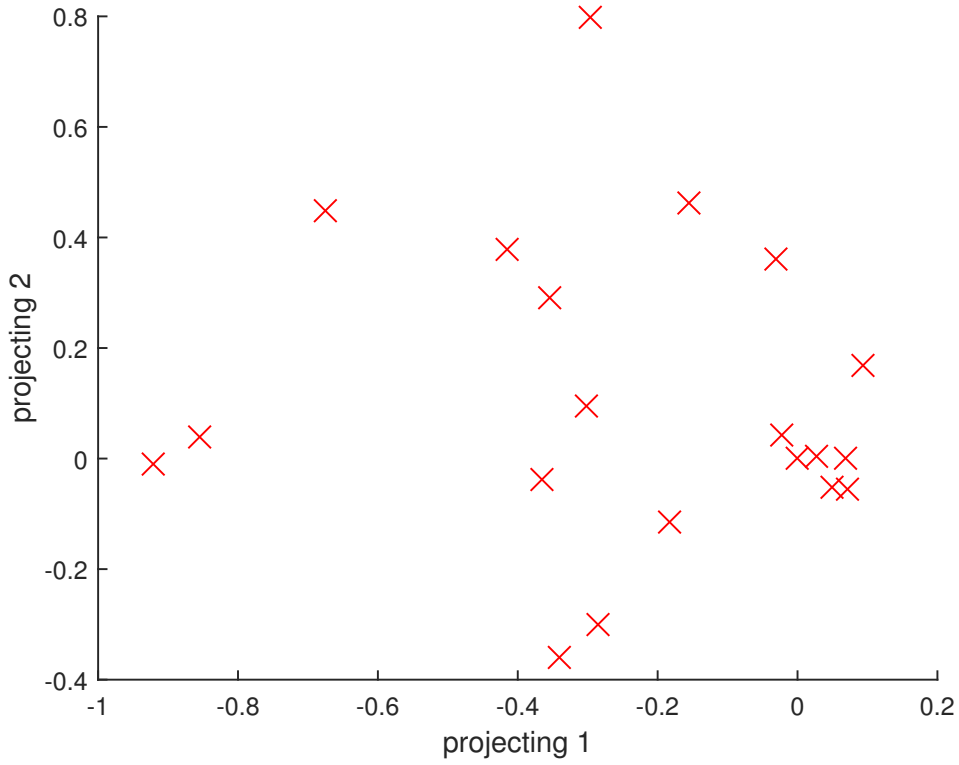
Once a flat region has been defined, the next step is to form triangles to describe this region. Since the region is almost flat, then without losing much information, the 3D points can be converted to be the 2D points, by projecting into this plane. An example of a number of 3D points in a flat region, and the corresponding 2D projections in this plane are shown in Figure 6.1a and 6.1b, respectively. To describe a 2D region, a common approach is by employing the technique Delaunay triangulation. An example of forming triangles to describe the region shown in Figure 6.1b by employing the technique Delaunay triangulation is shown in Figure 6.1c. To reduce the computational load, an alternative approach has been proposed. Detecting the boundary points of a region, and then forming triangles based on the boundary points to describe the region. An example of detecting the boundary points for the region shown in Figure 6.1b, and further forming triangles based on the detected boundary points are shown in Figure 6.1d and 6.1e, respectively. Comparing Figure 6.1c and 6.1e, it can be seen that the proposed method uses a much smaller number of triangles (i.e. just 4) to describe the region.

Mapping 2D Triangles

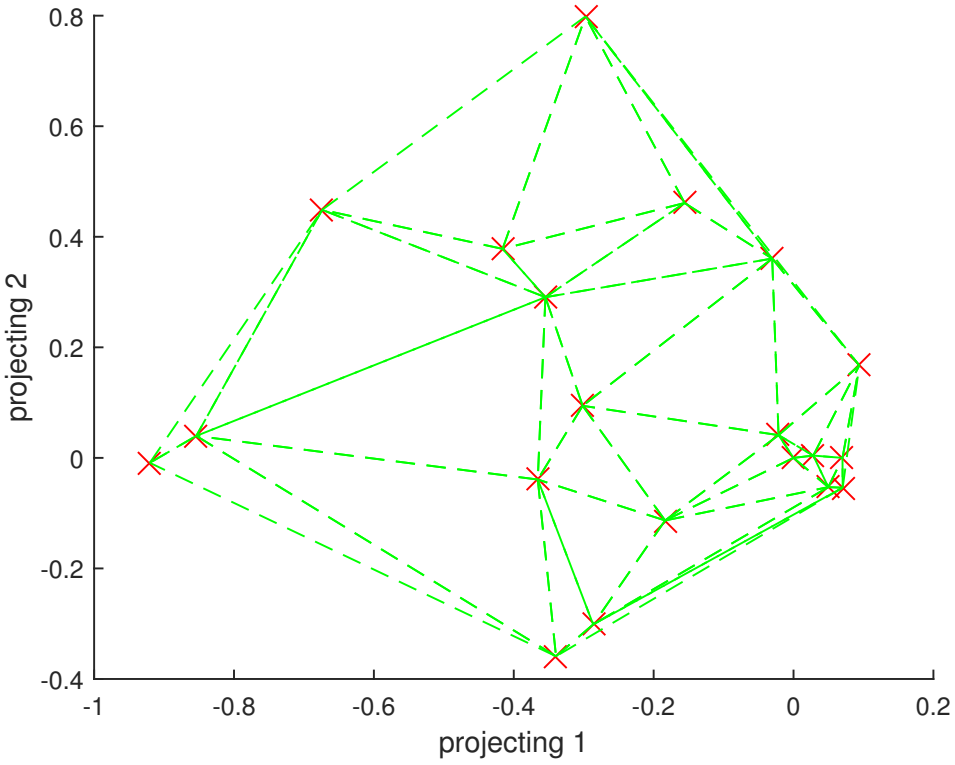
The last challenge left is how to map a 2D image triangle to a 3D space. Since the three 3D points can uniquely determine a plane, the corresponding mapping process is reduced to a 2D transformation between two planes. However, to determine a homography matrix $\mathbf{H}$, there needs to be at least 4 pairs of point correspondences; however there are only 3 pairs (i.e. the three triangle vertices defining the plane). Without losing any generalisation, the 4th pair of point correspondences can be found by using the triangle centre (i.e. the mean coordinate of the three triangle vertices). An example of mapping a 2D image triangle to the corresponding position in the plane determined by three 3D points is shown in Figure 6.2.


z
1
0.8
0.6
0.4
0.2
y
1
0.5
0
x
0
0.2
0.4
0.6
0.8
1

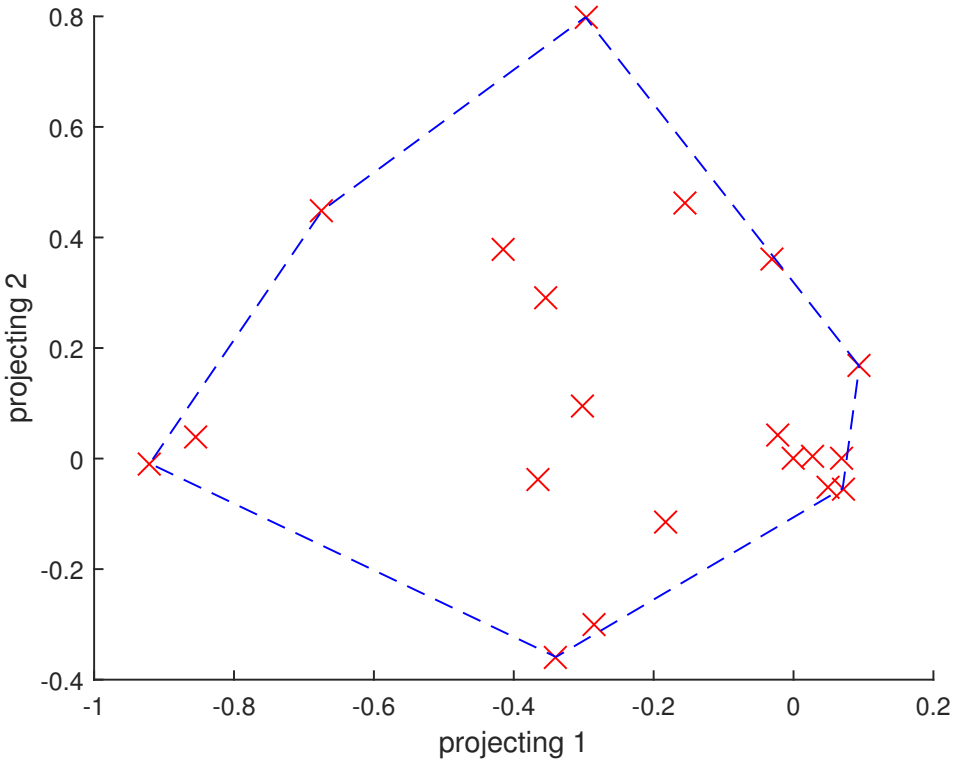
(a)


projecting 2
0.8
0.6
0.4
0.2
0
-0.2
-0.4
projecting 1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2

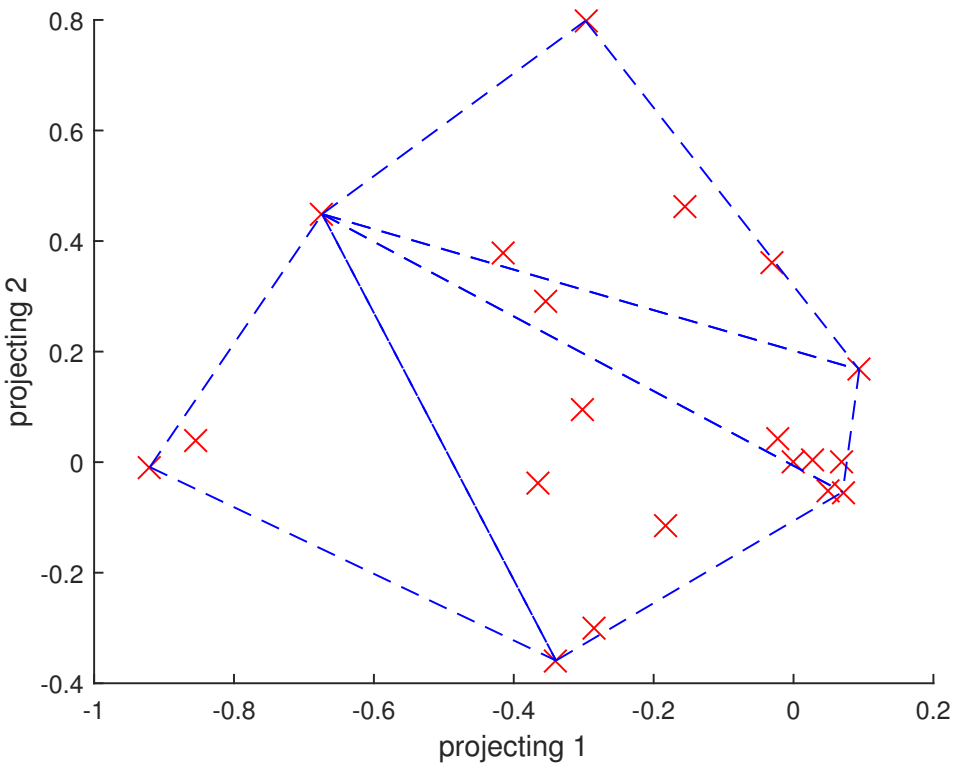
(b)


0.8
0.6
0.4
0.2
0
-0.2
-0.4
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
projecting 2
projecting 1

(c)


0.8
0.6
0.4
0.2
0
-0.2
-0.4
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
projecting 2
projecting 1

(d)



(e)

Figure 6.1: An example of forming triangles for a number of 3D points in a flat region: (a) a number of 3D points in a flat region (i.e. they almost lie on a common plane) (b) the projections of the 3D points in this common plane (c) the triangles formed by employing the technique Delaunay triangulation (d) the boundary points of this region (e) the triangles formed by employing the proposed method.

6.3 Experimental Evaluation

To evaluate the performance of the proposed method, four different sets of 2D images, namely, graf, bark, wall and vgm, were used to generate sparse point clouds, by the SfM technique. Each of the four sets contains six 2D images of the same scene, as shown in Figure 6.3, Figure 6.4, Figure 6.5 and Figure 6.6, respectively. Here, the application VisualSFM [300] were used, which is an existing implementation of the SfM technique. The generated sparse point clouds are plotted in Figure 6.7, Figure 6.8, Figure 6.9 and Figure 6.10, respectively. Further, dense point clouds were generated to improve the density of the corresponding sparse point clouds, as shown in Figure 6.11, Figure 6.12, Figure 6.13 and Figure 6.14, respectively. Here, the application CMVS [301] was used, which is based on the technique Clustering Views for Multi-view Stereo (CMVS) [250, 251]. Comparing the point clouds shown in Figure 6.7 and Figure 6.11, Figure 6.8 and Figure 6.12, Figure 6.9 and Figure 6.13, as well as Figure 6.10 and Figure 6.14, it can be seen that there is almost no improvement by employing

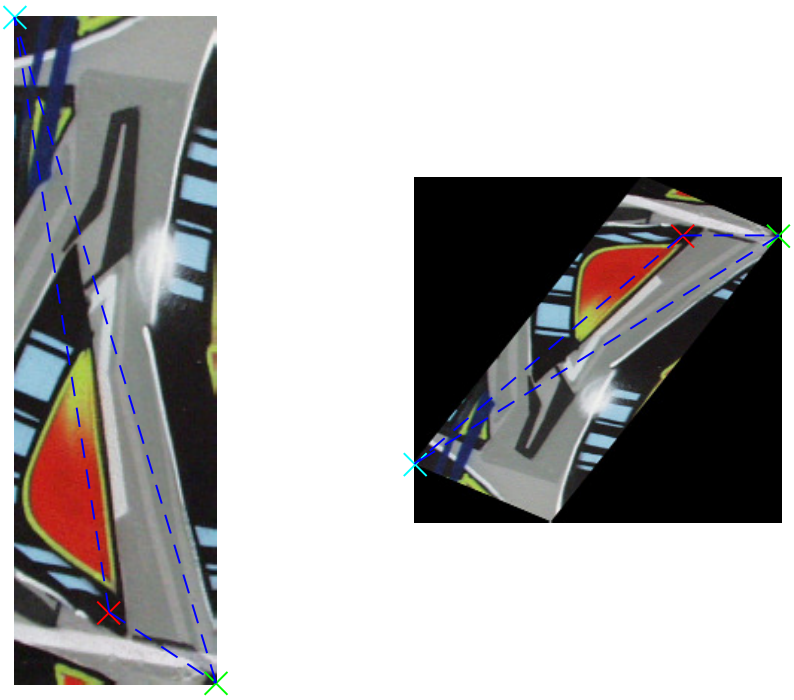


Figure 6.2: An example of mapping a 2D image triangle to the corresponding position in the plane determined by three 3D points.

the CMVS technique, this is because there is a limited number of images available to capture information. The numbers of 3D points available in the different sparse and dense point clouds are listed in Table 6.1. From Table 6.1, it can be seen that after applying the CMVS technique, there is only a slight increment in the number of 3D points. Finally, 3D reconstructions corresponding to the four different sets are shown in Figure 6.15, Figure 6.16, Figure 6.17 and Figure 6.18, respectively. They were achieved by employing the method proposed in this thesis, based on the sparse point clouds shown in Figure 6.7, Figure 6.8, Figure 6.9 and Figure 6.10, respectively. From Figure 6.15, Figure 6.16, Figure 6.17 and Figure 6.18, it can be seen that most of the details have been successfully reconstructed, which demonstrates the performance of the proposed method.

The most computational complexity part in this kind of 3D reconstruction process is mapping 2D image triangles to the corresponding positions in the 3D scene. Compared to the traditional methods that map small triangles one by one, the method proposed in this thesis uses large triangles (covering a number of small triangles), which are formed based on boundary points, for flat and almost flat regions. It results of a significant decrease in the number of 2D image triangles need to be mapped back. For a traditional method (i.e. the Delaunay triangulation technique) and the proposed

Table 6.1: Numerical comparison of the performance of sparse and dense point clouds in terms of the number of 3D points in point cloud, as well as the computational complexity of the Delaunay triangulation technique and the proposed method in terms of the number of 2D image triangles need to map, for the four different image sets: graf, bark, wall and vgm.

	no. of 3D points	
	sparse point cloud	dense point cloud
graf	3,114	3,202
bark	5,894	5,968
wall	3,801	3,908
vgm	2,530	2,587

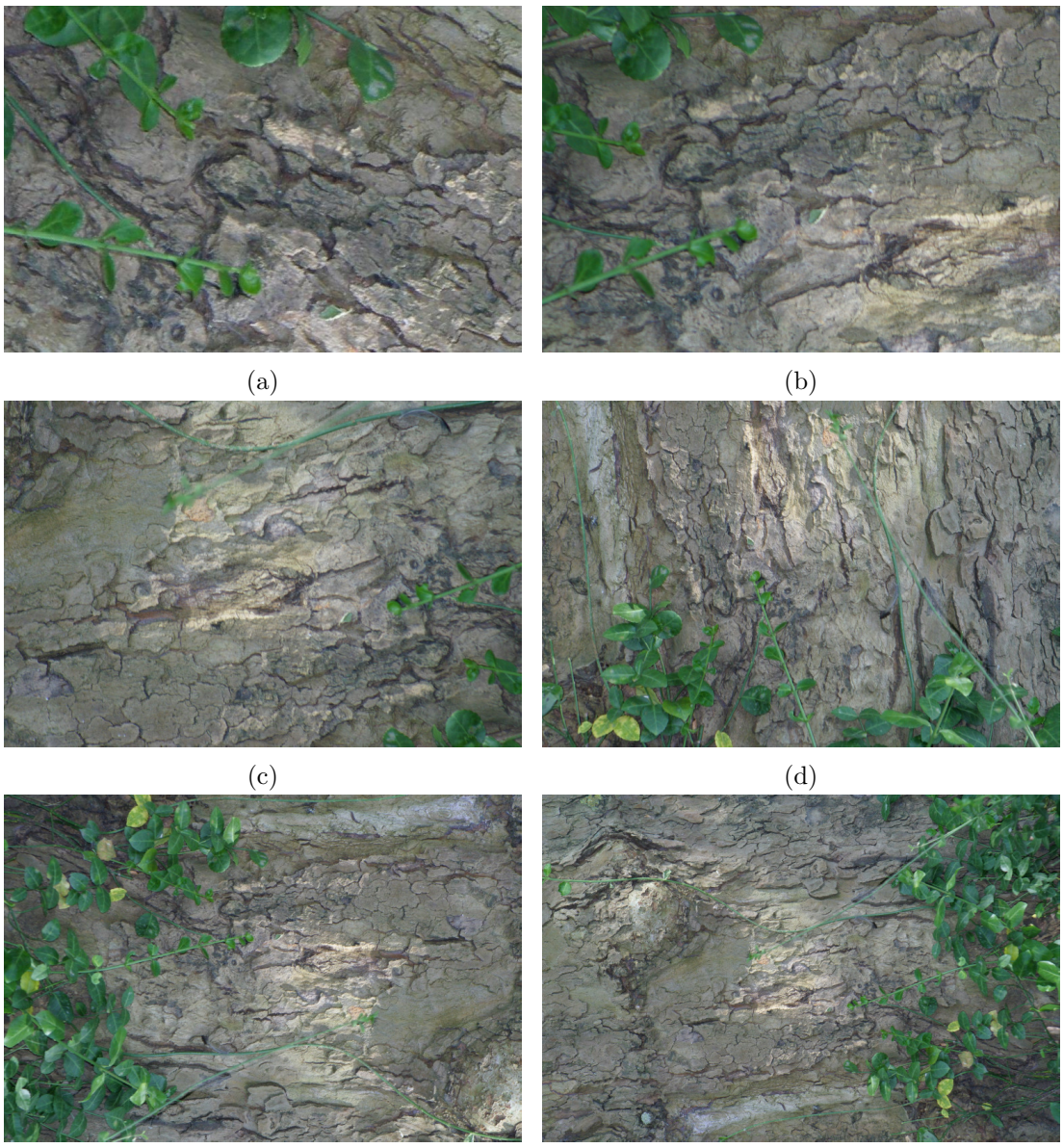
	no. of 2D triangles	
	Delaunay triangulation	proposed
graf	12,657	992
bark	31,112	918
wall	17,899	1,170
vgm	14,651	446

method, the numbers of 2D image triangles need to map are listed in Table 6.1. From Table 6.1, it can be seen that for each of the four different image sets, the number of 2D image triangles need to map for the proposed method is much smaller, compared to that for the Delaunay triangulation technique. As a result, the proposed method can outperform the existing techniques in the computational speed, while the corresponding reconstruction performance is still comparable.

To sum up, when there is only a small number of 2D images available to capture the information of a 3D scene, a sparse point cloud reconstructed by the SfM technique is sufficient, since the further reconstruction process for a dense point cloud by the CMVS technique would only improve a very small degree of the density of the point cloud. Compared to the the existing 3D reconstruction techniques based on mapping 2D image triangles, the proposed method is better at handling flat or almost flat regions in 3D scenes. As a result, the proposed method is especially suitable for the situations where the target 3D scene is flat or almost flat, or it contains a large degree of flat or almost flat regions.

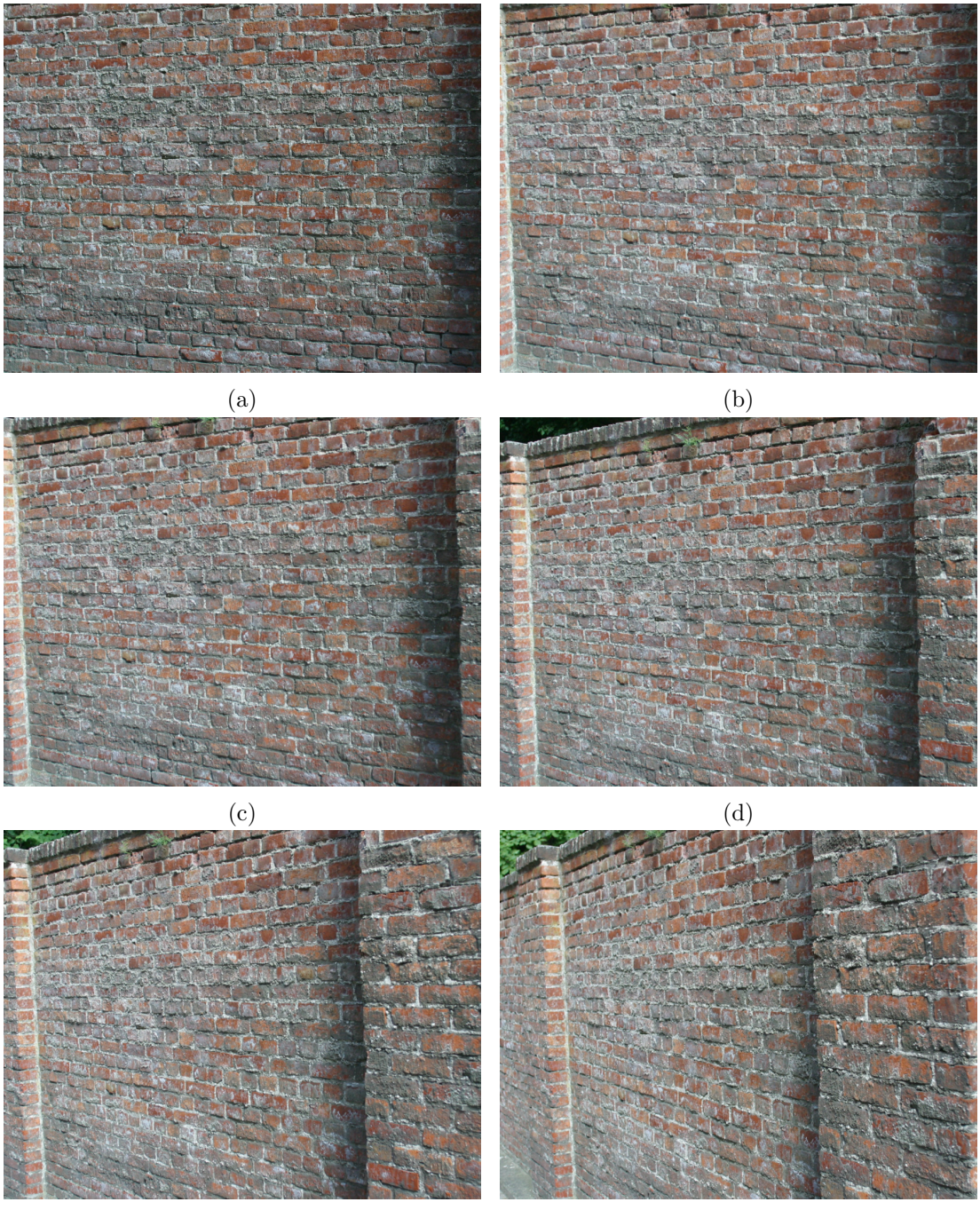
(a) (b) (c) (d) (e) (f)

Figure 6.3: The graf set of six 2D images of the same scene for generating a sparse point cloud, by the SfM technique.



(a) (b) (c) (d) (e) (f)

Figure 6.4: The bark set of six 2D images of the same scene for generating a sparse point cloud, by the SfM technique.

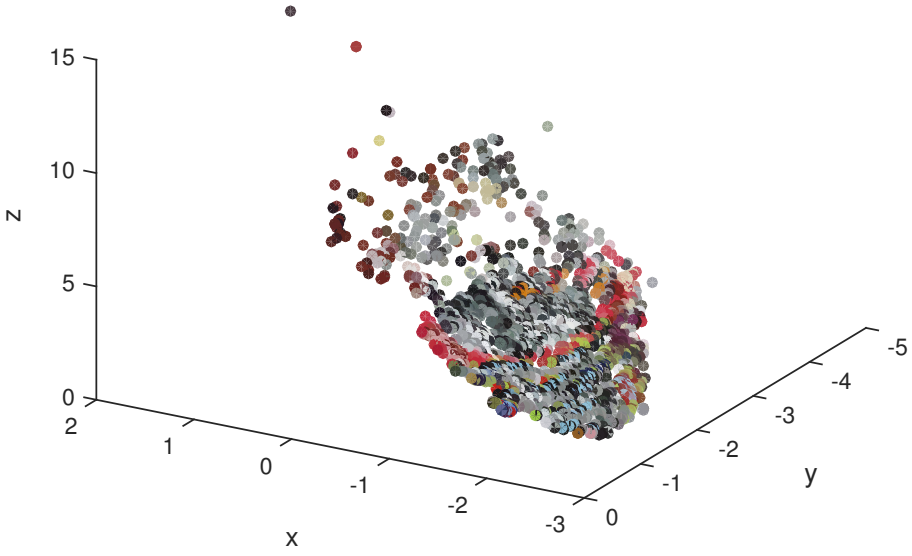


(a) (b) (c) (d) (e) (f)

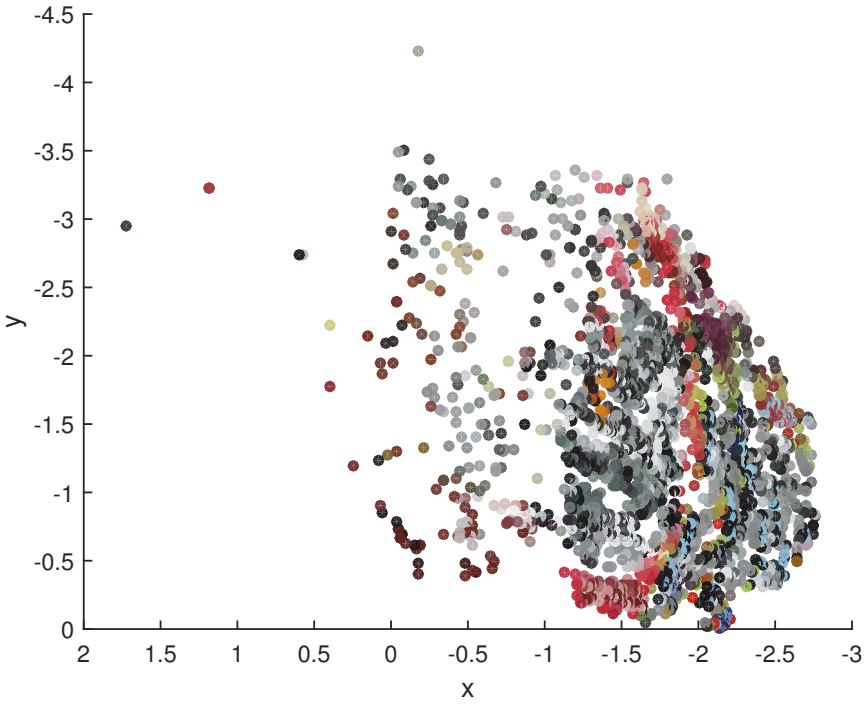
Figure 6.5: The wall set of six 2D images of the same scene for generating a sparse point cloud, by the SfM technique.

(a) (b) (c) (d) (e) (f)

Figure 6.6: The vgm set of six 2D images of the same scene for generating a sparse point cloud, by the SfM technique.

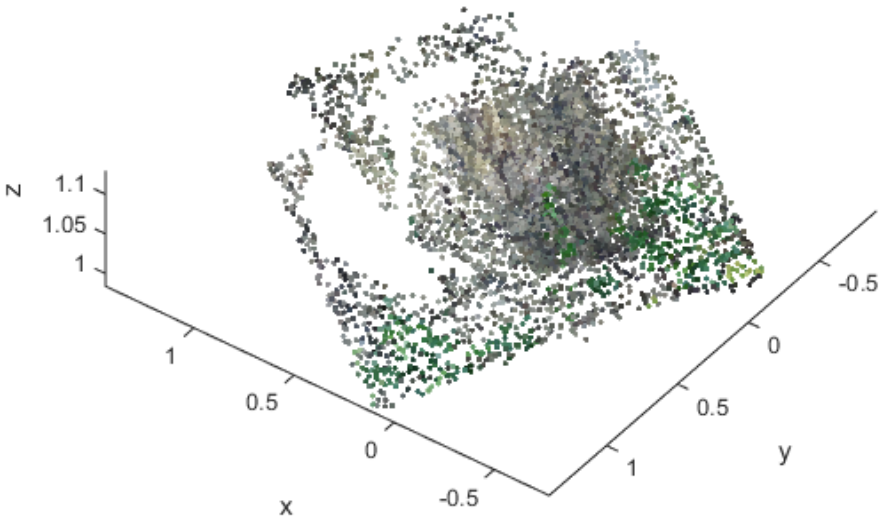


(a)

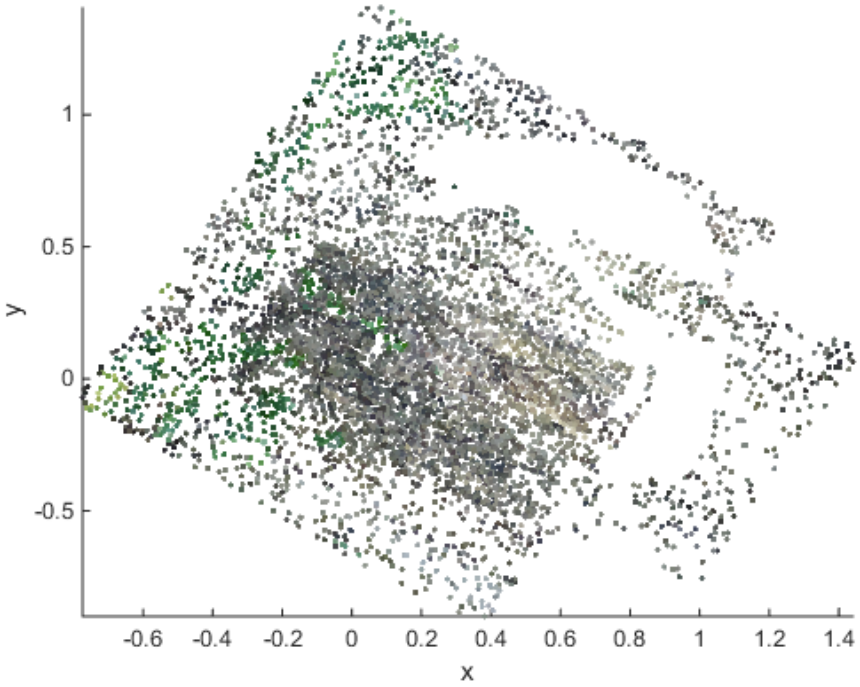


(b)

Figure 6.7: A sparse point cloud generated by employing the application VisualSFM [300], based on the six 2D image shown in Figure 6.3.

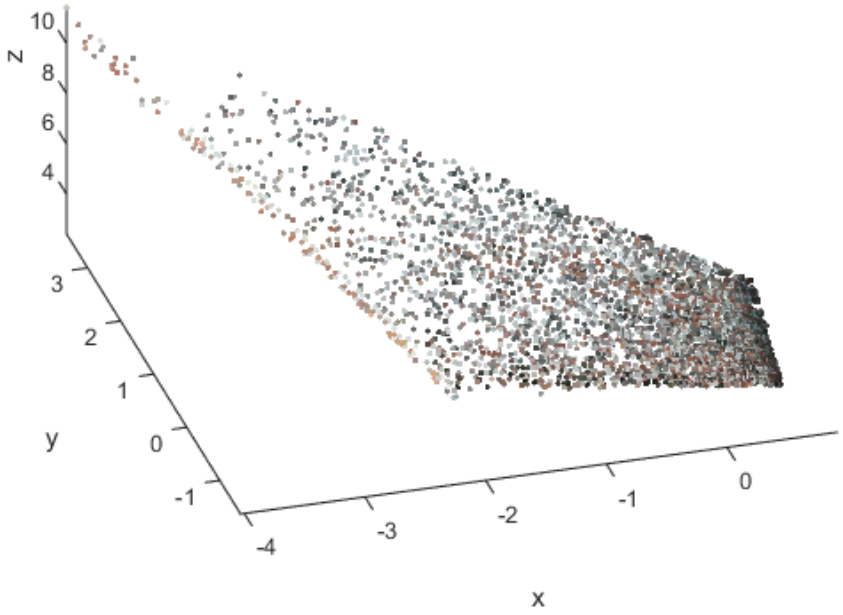


(a)

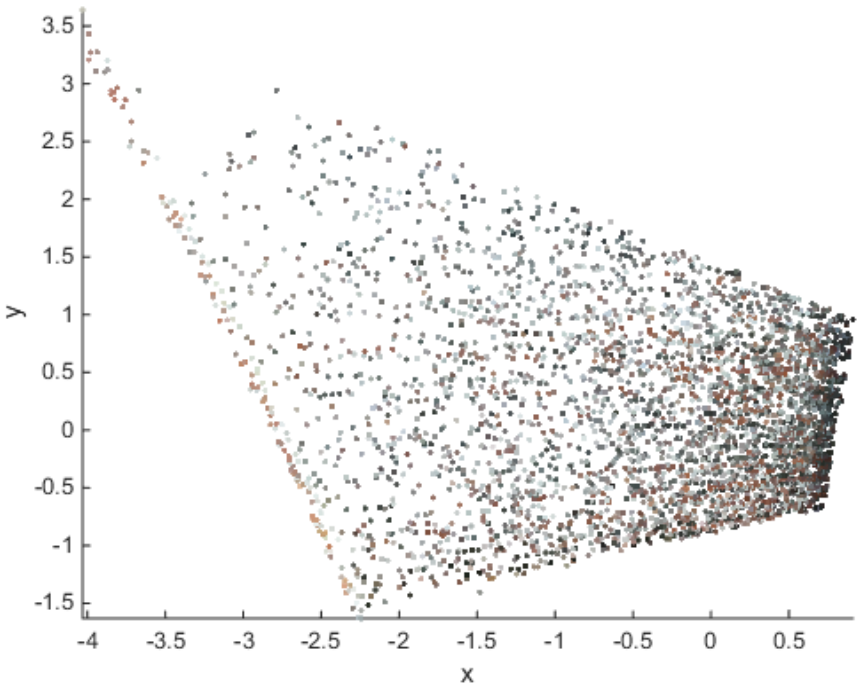


(b)

Figure 6.8: A sparse point cloud generated by employing the application VisualSFM [300], based on the six 2D image shown in Figure 6.4.

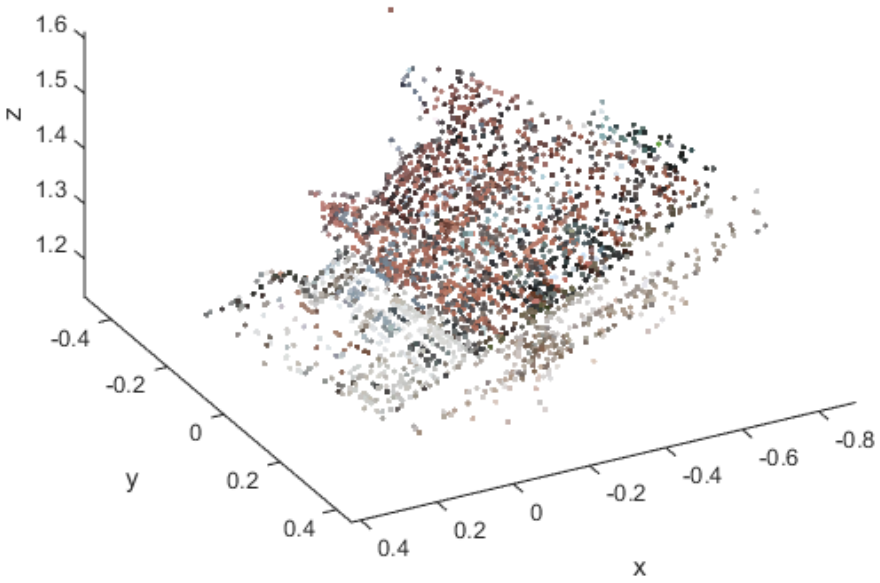


(a)

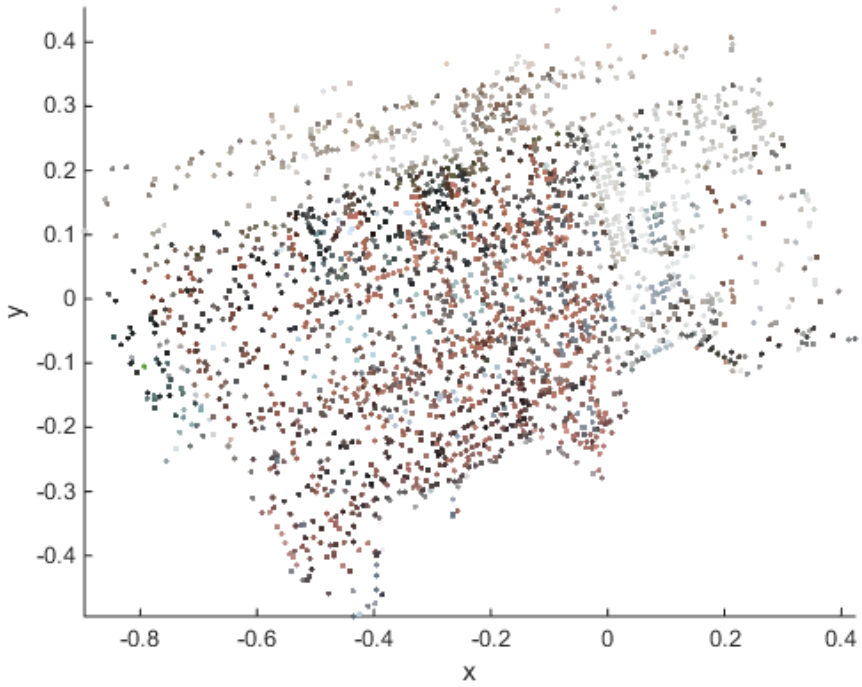


(b)

Figure 6.9: A sparse point cloud generated by employing the application VisualSFM [300], based on the six 2D image shown in Figure 6.5.

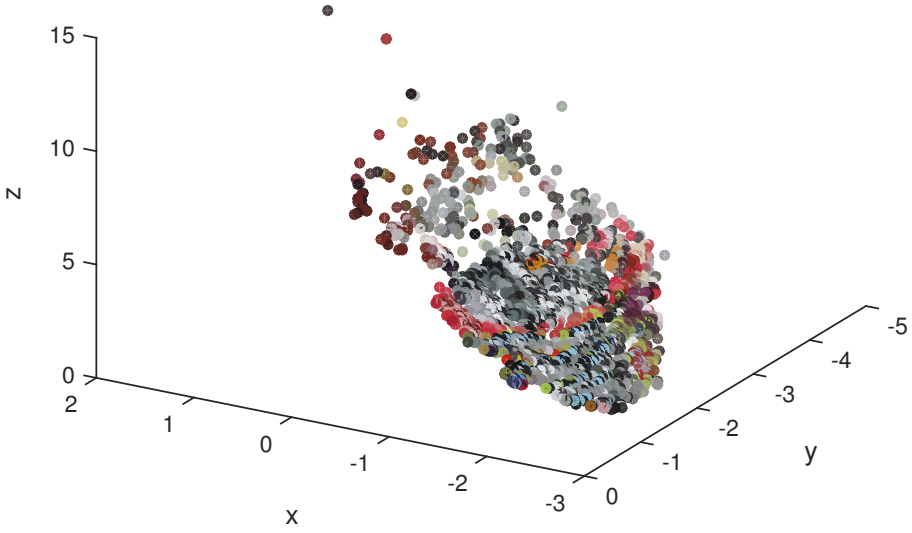


(a)

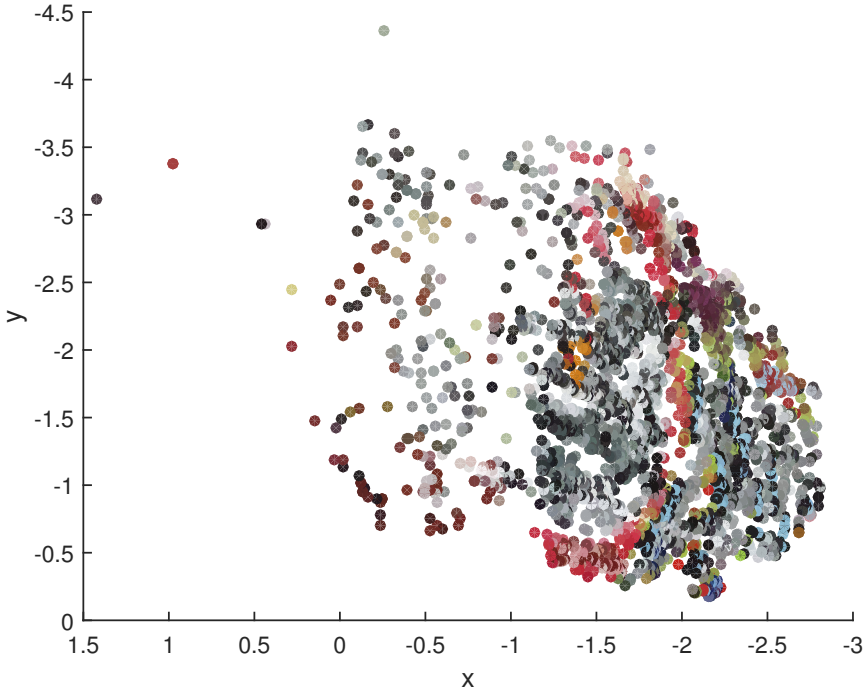


(b)

Figure 6.10: A sparse point cloud generated by employing the application VisualSFM [300], based on the six 2D image shown in Figure 6.6.

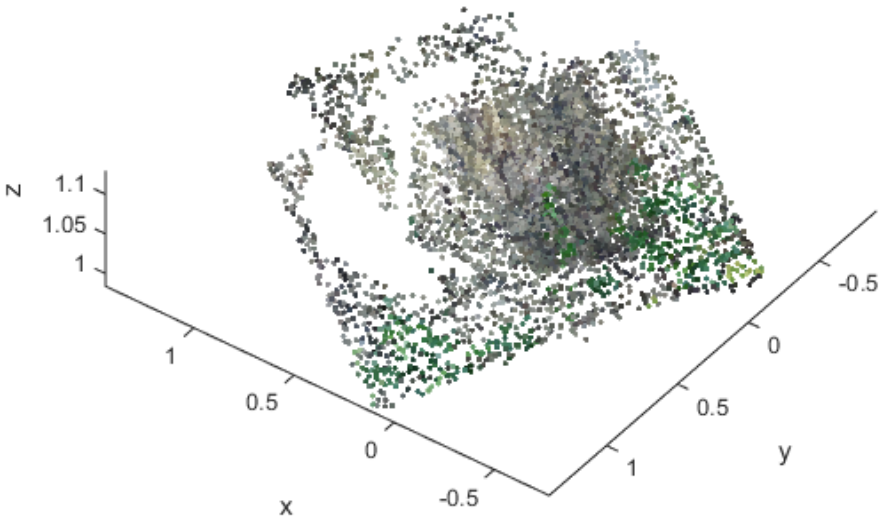


(a)

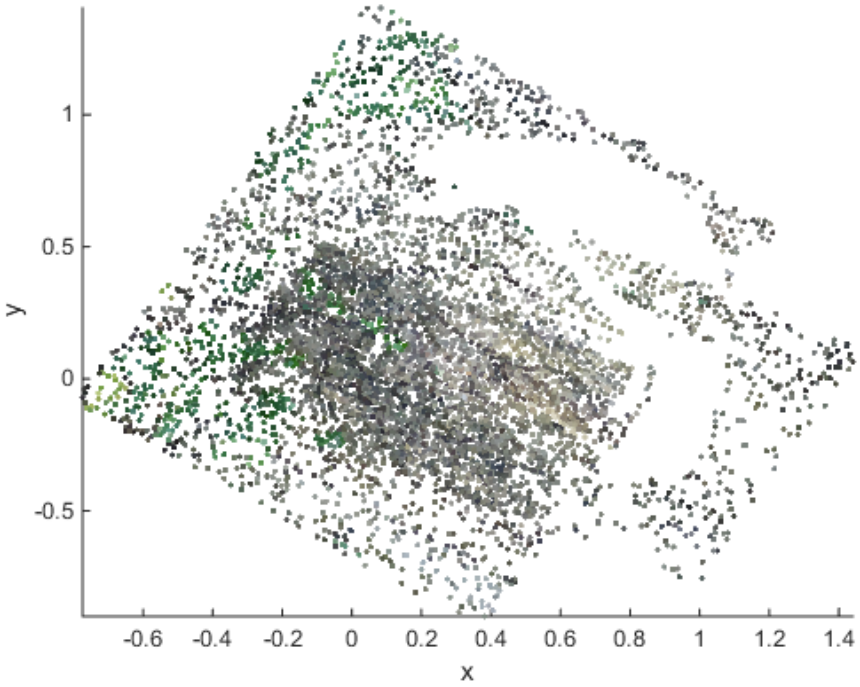


(b)

Figure 6.11: A dense point cloud generated by employing the application CMVS [301], based on the sparse point cloud shown in Figure 6.7.

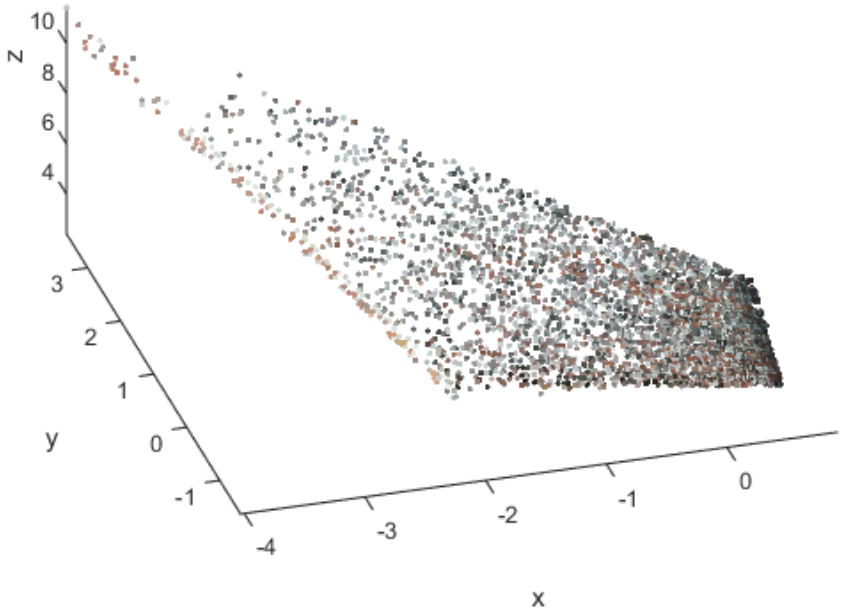


(a)

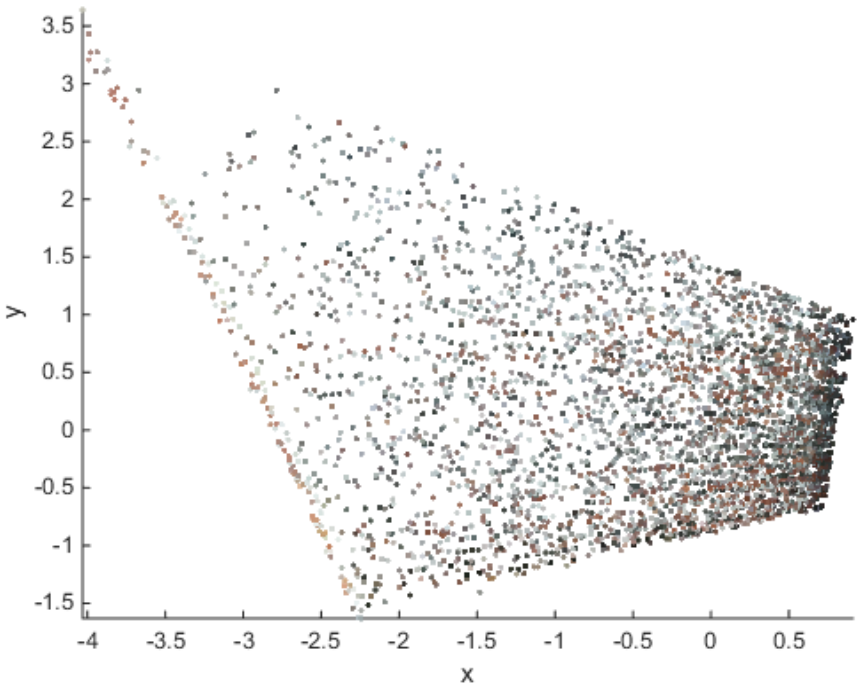


(b)

Figure 6.12: A dense point cloud generated by employing the application CMVS [301], based on the sparse point cloud shown in Figure 6.8.

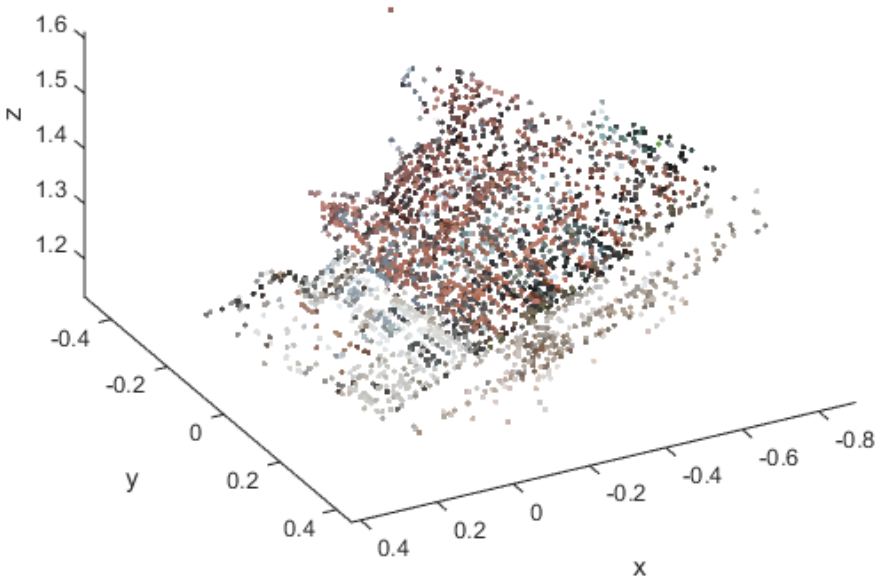


(a)

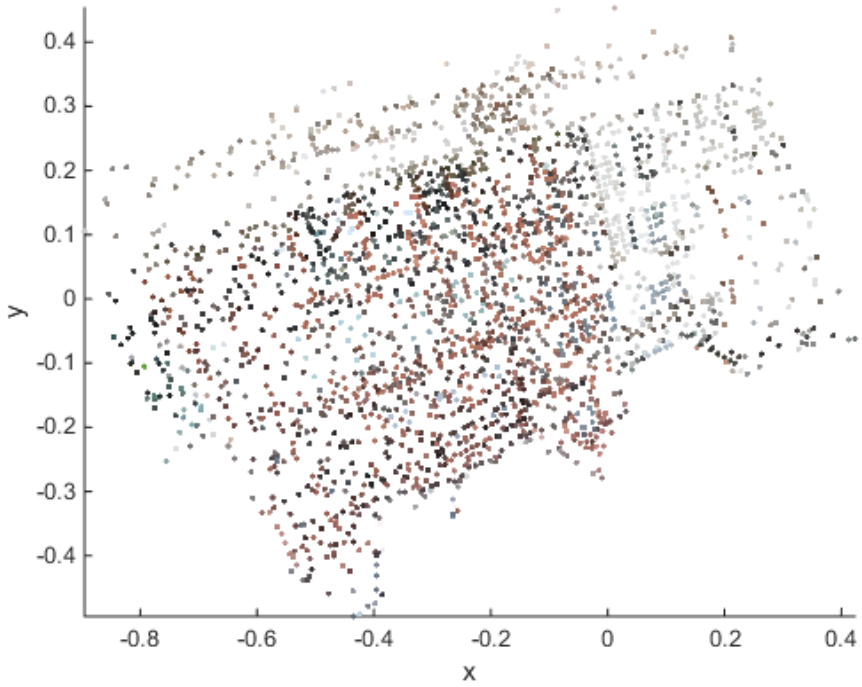


(b)

Figure 6.13: A dense point cloud generated by employing the application CMVS [301], based on the sparse point cloud shown in Figure 6.9.

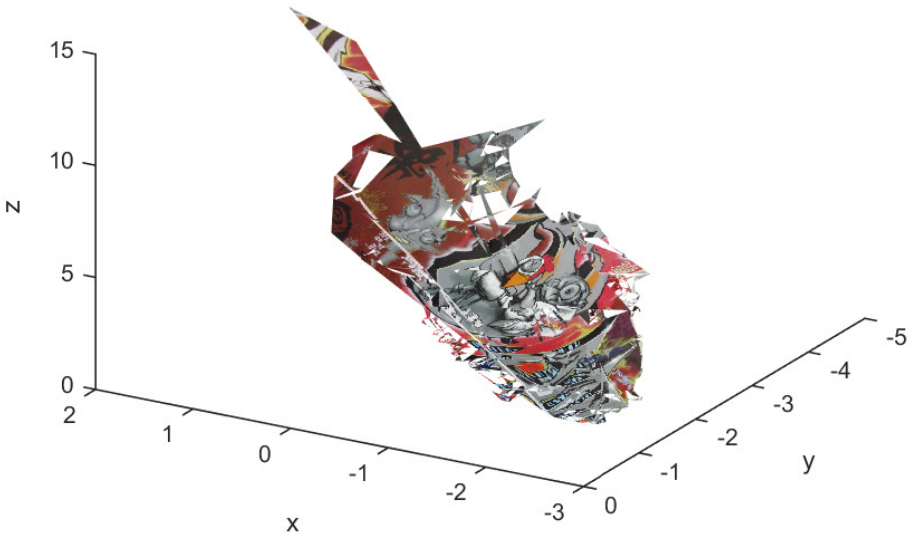


(a)

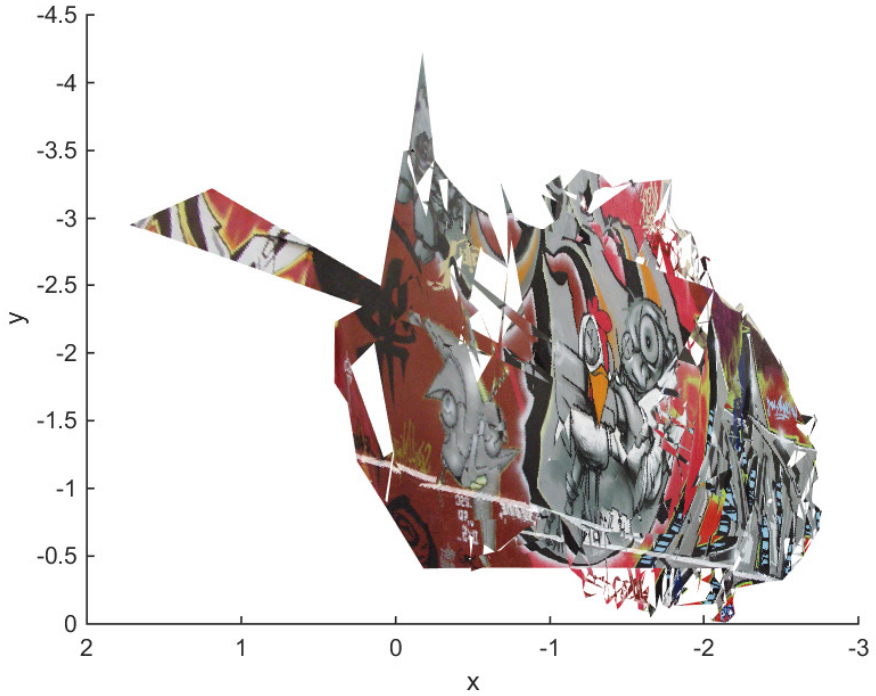


(b)

Figure 6.14: A dense point cloud generated by employing the application CMVS [301], based on the sparse point cloud shown in Figure 6.10.

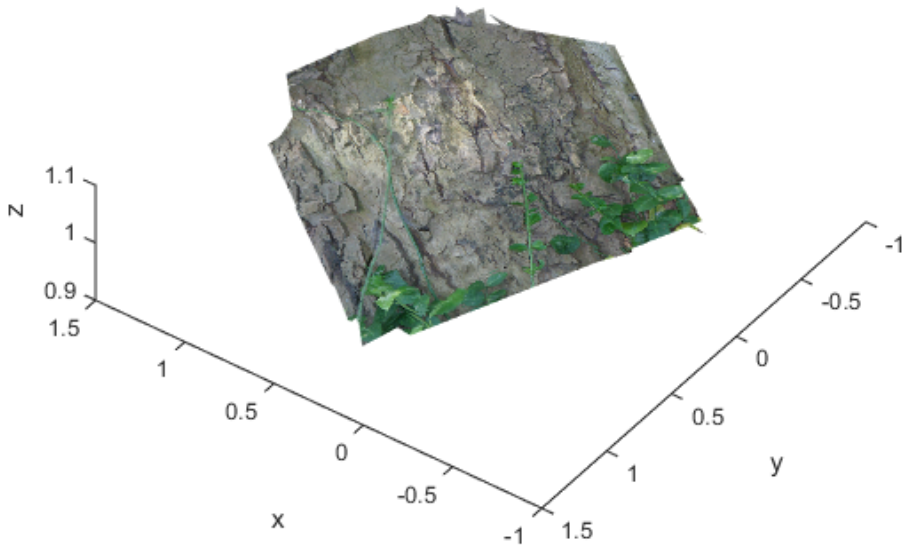


(a)

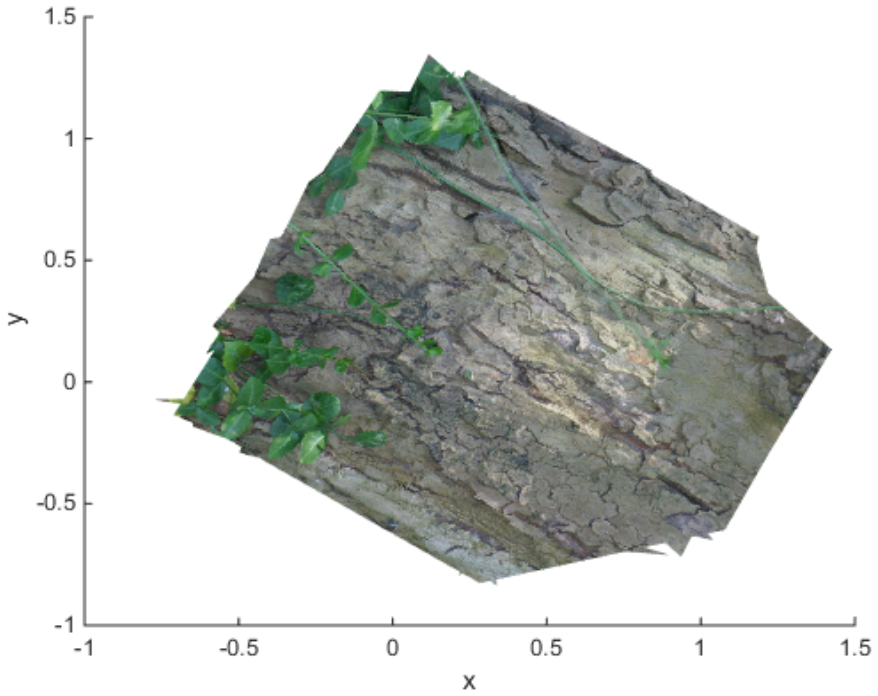


(b)

Figure 6.15: The 3D reconstruction result by employing the proposed method, based on the sparse point cloud shown in Figure 6.7.

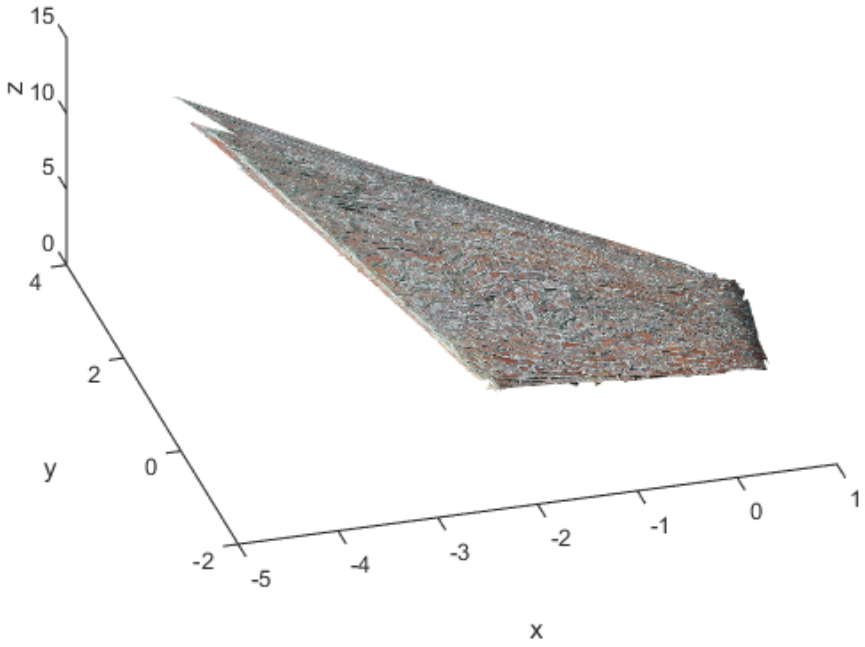


(a)

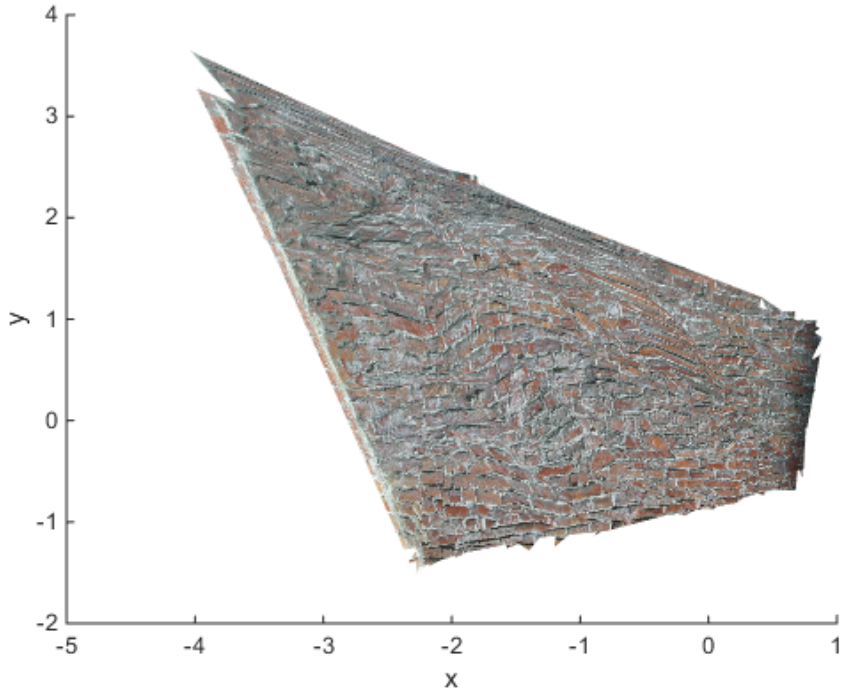


(b)

Figure 6.16: The 3D reconstruction result by employing the proposed method, based on the sparse point cloud shown in Figure 6.8.

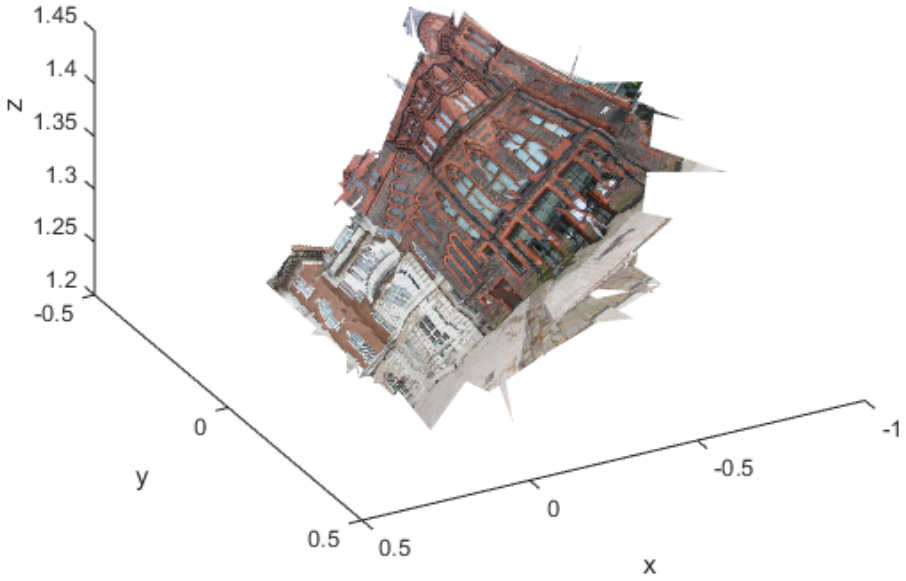


(a)

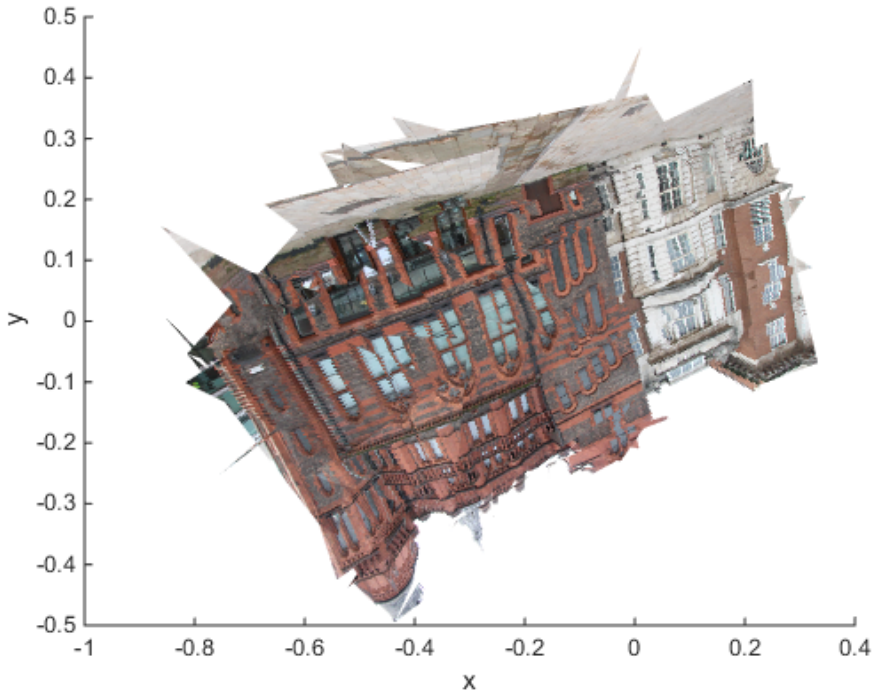


(b)

Figure 6.17: The 3D reconstruction result by employing the proposed method, based on the sparse point cloud shown in Figure 6.9.



(a)



(b)

Figure 6.18: The 3D reconstruction result by employing the proposed method, based on the sparse point cloud shown in Figure 6.10.

Chapter 7

Conclusions & Future Work

7.1 Summary

In this thesis, data processing techniques have been developed for three different but related applications: embedding learning for classification, fusing low bit depth images, and 3D reconstruction from 2D images.

For the applications of embedding learning for classification, a method for manifold embedding has been proposed to facilitate the task of multiclass classification. The proposed method is based on binary classification: treat each class individually so that the salient features for each class can be uniquely identified. This binary data embedding framework is based on two processing stages: pre-processing and embedding computation. In the embedding computation stage, the concepts of friend closeness and enemy dispersion are utilised to generate adaptive measures of the intraclass and interclass information. These measures can be seen as a generalisation of the Fisher criterion [50], and additionally incorporating the local neighbour information into the calculations of the corresponding proximity matrices. The presented experimental results demonstrate the effectiveness of the proposed method by comparison with seven existing techniques for embedding learning (both unsupervised and supervised), using four face databases and two text datasets. It has been showed that the method proposed in this thesis performed better than these existing techniques, especially in the situation where only a small number of data samples were available for training. The approach is relevant to human supervised computer identification for security and other relevant applications. Using such processing may reduce the workload of a human operator and improve identification performance.

For the applications of fusing low bit depth images, it has been shown that the Phase Correlation method can be successfully applied to register 2-bit depth images of the same scene that are created by a similarity distortion, without any modification, or pre and/or post-processing. The information in each 2-bit image could be acquired at a temporally finer resolution, potentially operating at a higher frame rate, coping with rapid micro-UAV movements and producing smoother video to the operator, without

extra overheads of compression on-board the UAV. These computational overheads need to be minimised due to the Size, Weight and Power - Cost (SWaP-C) requirements of small and micro-scale UAVs. Fusing the aligned 2-bit depth images together results in an informative image, this has been demonstrated by aligning and then fusing nine 2-bit depth images of the same scene which obtains comparable detail to the corresponding 8-bit depth image. It is concluded that with these techniques it is possible for a UAV operator/computer on the ground to reconstruct the imagery from a UAV that is just transmitting the two most significant bits of each image frame without further on-board processing.

For the application of reconstructing 3D scene from 2D images, a method has been proposed to handle the dense reconstruction based on a sparse 3D point cloud generated by the structure from motion technique. Instead of improving the density of the 3D point cloud, it achieves reconstruction by mapping 2D image triangles back to the corresponding positions (i.e. triangles formed by three 3D points) in the 3D scene. Compared to the existing methods that use a similar approach, the proposed method differs by using large triangles to replace a number of small triangles for flat and almost flat areas. The proposed method aims to reduce the number of 2D image triangles that need to be mapped back to the 3D scene, so that the computational load can be reduced. To describe a flat or almost flat area, it has been demonstrated that a minimum number of triangles can be achieved by forming triangles based on utilising the boundary points of this region. From the experimental results, it can be concluded that the proposed method can achieve better reconstruction details when compared to the spare and dense reconstruction methods by 3D point cloud; while it can achieve a comparable reconstruction result while the computational speed is several times faster when compared to the existing methods that use a similar approach but map every 2D image triangles.

7.2 Future Work

For feature based image registration, sometimes the existing algorithms fail because there are quite similar features appearing in the same image, which results in the constructed feature descriptors being too similar. One possible solution to this is to use embedding learning. Since the spatial location information is not utilised when constructing feature descriptors, it can be used by the embedding learning to project the feature descriptors into a new feature space, to improve the discrimination of the feature points thereby improve association of features. Embedding learning is a general technique for high dimensional data, such as images, and the full range of its applications have not yet been explored.

Appendix A

Feature Based Weight Matrix

The pairwise weight matrix $\mathbf{W}$ employed in LPP, OLPP or any other algorithms proposed based on manifold learning and spectral analysis is mainly based on a specific similarity or dissimilarity measure between all the paris of the data samples in the original set. In the literature, there are a wide set of simple and efficient techniques have been developed to define w_{ij}, which is a similarity or dissimilarity measure between the data samples $\mathbf{x}_i$ and $\mathbf{x}_j$.

Simple Minded

It uses only the consonant value 1 to capture the feature based information, which indicates that the two data samples are connected. It is defined as:

$$w_{ij} = 1 \quad \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are connected} \tag{A.1}$$

There are various more complex measures, which are proposed based on dot product, polynomial kernel or correlation coefficient, and some commonly used ones are listed as follows:

Cosine Similarity

It is a measure of similarity between two data samples of an inner product space, which measures the cosine of the angle between them [302]. It is defined according to:

$$w_{ij} = \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2} \tag{A.2}$$

Polynomial Kernel

It is a kernel function that commonly used with kernelised models, such as SVMs. It allows the learning of nonlinear models, by representing the similarity of data samples in a feature space over polynomials of the original feature space, and it is computed as:

$$w_{ij} = (\mathbf{x}_i^T \mathbf{x}_j + c)^d \tag{A.3}$$

where $c \geq 0$ is a constant that tradea off the influence of higher oder versus lower order terms in the polynomial, and d indicates the polynomial degree [303].

Tanimoto Similarity

It is a derived distance function that measures a similarity ratio between two data samples [304–306], and it is defined according to:

$$w_{ij} = \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\|_2^2 \|\mathbf{x}_j\|_2^2 - \mathbf{x}_i^T \mathbf{x}_j} \tag{A.4}$$

Gaussian Weight

It is also known as heat kernel, and it is the most popular measure [44, 45], as it can adequately characterise the relation included in the features [103]. Additionally, it captures the interactions between data samples, which may potentially discovery the nonlinear structures included in the data set [14]. It is defined as:

$$w_{ij} = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma}) \tag{A.5}$$

where $\sigma > 0$ determines the width of the Gaussian kernel.

Alternative Weight

It is suggested to be used for the data set, where the data samples from different classes are close in the original high dimensional space [45], and it is computed as:

$$w_{ij} = \frac{1}{\tau + \frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\|\mathbf{x}_i\|_2^2 + \|\mathbf{x}_j\|_2^2}} \tag{A.6}$$

It is stated in [45] that the different values chosen for τ would result of a similar final performance.

Pearson Correlation Coefficient

It is a measure of the linear correlation between two data samples, to represent the degree of linear dependence between two variables, whose result is in a range between $+1$ and -1 inclusive. It is computed according to:

$$w_{ij} = \frac{\sum_{p=1}^{d} \left(x_{ip} - \frac{1}{d} \sum_{q=1}^{d} x_{iq} \right) \left(x_{jp} - \frac{1}{d} \sum_{q=1}^{d} x_{iq} \right)}{\sqrt{\sum_{p=1}^{d} \left(x_{ip} - \frac{1}{d} \sum_{q=1}^{d} x_{iq} \right)^2} \sqrt{\sum_{p=1}^{d} \left(x_{jp} - \frac{1}{d} \sum_{q=1}^{d} x_{jq} \right)^2}} \tag{A.7}$$

A.1 LLE Style Weight

Here, the approach of how to find out the optimal reconstruction weights of LLE [13] is examined in details. The optimal reconstruction weights of LLE is based on the assumption that each data sample can be reconstructed by a linear combination of its local neighbouring ones. Then it implies that each data sample and its local neighbours do form a linear subspace. Thus, first it is necessary to use k-NN to establish some local neighbours for each data sample, so that each data same is conformed or adapted.

Assume that each data sample in the original set $\mathbf{x}_i$ $(i = 1, \cdots, n)$ can be reconstructed by a linear combination of its k-NN ones $\mathbf{x}_j$ $(j = 1, \cdots, k)$, where the corresponding weights w_{ij} $(j = 1, \cdots, k)$ sum to 1. Then, the corresponding reconstruction error ϵ can be computed as:

$$\begin{aligned}
\epsilon &= \left\| \mathbf{x}_i - \sum_{j=1}^{k} w_{ij}\mathbf{x}_j \right\|_2^2 \\
&= \left\| \sum_{j=1}^{k} w_{ij}(\mathbf{x}_i - \mathbf{x}_j) \right\|_2^2 \\
&= \|\mathbf{w}_i \mathbf{X}_i\|_2^2 \\
&= \mathbf{w}_i \mathbf{X}_i \mathbf{X}_i^T \mathbf{w}_i^T
\end{aligned} \tag{A.8}$$

where $\mathbf{w}_i = [w_{i1}, w_{i2}, \cdots, w_{ik}]$ is a $1 \times k$ matrix that includes all the reconstruction k weights, and $\mathbf{X}_i = [\mathbf{x}_i - \mathbf{x}_1, \mathbf{x}_i - \mathbf{x}_2, \cdots, \mathbf{x}_i - \mathbf{x}_k]^T$ is a $k \times d$ matrix that represents all the differences between the data sample $\mathbf{x}_i$ and each of its k-NN ones. Then, the optimal weights can be obtained by optimising the following minimisation problem:

$$\underset{\mathbf{w}_i \mathbf{1}_{k\times 1}=1}{\arg\min} \frac{1}{2} \mathbf{w}_i \mathbf{X}_i \mathbf{X}_i^T \mathbf{w}_i^T \tag{A.9}$$

where $\mathbf{X}_i \mathbf{X}_i^T$ is the $k \times k$ local covariance matrix associated with the data sample $\mathbf{x}_i$. By employing the Lagrangian multiplier λ_i, the above optimisation problem can be reformulated as:

$$L(\mathbf{w}_i, \lambda_i) = \frac{1}{2} \mathbf{w}_i \mathbf{X}_i \mathbf{X}_i^T \mathbf{w}_i^T - \lambda_i (\mathbf{w}_i \mathbf{1}_{k\times 1} - 1) \tag{A.10}$$

Take the derivatives each with respect to $\mathbf{w}_i$ and λ_i, as:

$$\begin{aligned}
\frac{\partial L(\mathbf{w}_i, \lambda_i)}{\partial \mathbf{w}_i} &= \frac{1}{2}(\mathbf{X}_i \mathbf{X}_i^T \mathbf{w}_i^T + \mathbf{X}_i \mathbf{X}_i^T \mathbf{w}_i^T) - \lambda_i \mathbf{1}_{k\times 1} \\
&= \mathbf{X}_i \mathbf{X}_i^T \mathbf{w}_i^T - \lambda_i \mathbf{1}_{k\times 1}
\end{aligned} \tag{A.11}$$

$$\frac{\partial L(\mathbf{w}_i, \lambda_i)}{\partial \lambda_i} = \mathbf{w}_i \mathbf{1}_{k\times 1} - 1 \tag{A.12}$$

Set (A.11) to $\mathbf{0}$, then the following equation can be obtained:

$$\mathbf{X}_i \mathbf{X}_i^T \mathbf{w}_i^T = \lambda_i \mathbf{1}_{k\times 1} \tag{A.13}$$

Thus, $\mathbf{w}_i$ can be obtained, as:

$$\mathbf{w}_i = \lambda_i \mathbf{1}_{1\times k} \left(\mathbf{X}_i \mathbf{X}_i^T\right)^{-1} \tag{A.14}$$

Similarly, set (A.12) to 0, and then couple the resultant equation with (A.14), λ_i can be found, as:

$$\lambda_i = \left(\mathbf{1}_{1\times k} \left(\mathbf{X}_i \mathbf{X}_i^T\right)^{-1} \mathbf{1}_{k\times 1}\right)^{-1} \tag{A.15}$$

Therefore, the optimal reconstruction weights $\mathbf{w}_i$ is obtained [307], as:

$$\mathbf{w}_i = \left(\mathbf{1}_{1\times k} \left(\mathbf{X}_i \mathbf{X}_i^T\right)^{-1} \mathbf{1}_{k\times 1}\right)^{-1} \mathbf{1}_{1\times k} \left(\mathbf{X}_i \mathbf{X}_i^T\right)^{-1} \tag{A.16}$$

Appendix B

Data Pre-Processing Techniques

Data pre-processing techniques (e.g. data normalisation) are widely employed in many areas of science and engineering, such as machine learning and data mining. Because they tend to eliminate irrelevant and/or redundant information that is present in a set of data samples, which results of enhancing the knowledge discovery ability of many existing algorithm. As a result, they can potentially improve the final performance of a method [263], and may also reduce the computational complexity of most of the algorithms proposed for machine learning and data mining, due to a large reduction in the dimensionality of an input set of data samples. In the literature, a wide set of such techniques have been proposed, which are simple and efficient [14, 263]. They can be divided into two different categories: one is simple normalisation; the other is dimensionality reduction, which applies simple normalisation first, and sequentially reduces the dimensionality.

B.1 Simple Normalisation

Usually, simple normalisation is the standard first stage in the process of data pre-processing. There are many different possible approaches, and three commonly used ones are listed as follows.

Rescaling

It rescales a data sample along each of its dimension, to make it become a unit 2-norm vector. Then, the training and test data samples $\mathbf{x}_i$ $(i = 1, \ldots, n)$ and $\widetilde{\mathbf{x}}_i$ $(i = 1, \ldots, m)$ are each rescaled according to:

$$\mathbf{x}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2} \tag{B.1}$$

and

$$\widetilde{\mathbf{x}}_i = \frac{\widetilde{\mathbf{x}}_i}{\|\widetilde{\mathbf{x}}_i\|_2} \tag{B.2}$$

respectively, where $\|\cdot\|_2$ donates the 2-norm or Euclidean norm.

Centring

This centres a set of data samples by subtracting the mean of each dimension from the corresponding dimension of each data sample, to make each dimension of the set possess a mean value of zero. Then, the training and test data samples $\mathbf{x}_i$ $(i = 1, \ldots, n)$ and $\widetilde{\mathbf{x}}_i$ $(i = 1, \ldots, m)$ are each centred according to:

$$\mathbf{x}_i = \mathbf{x}_i - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \tag{B.3}$$

and

$$\widetilde{\mathbf{x}}_i = \widetilde{\mathbf{x}}_i - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \tag{B.4}$$

respectively.

Standardisation

This standardises a set of data samples by centring each data sample first, and then dividing each of its dimension by the corresponding standard deviation, so that each dimension of the original set has the property of zero mean and unit variance. Then, the training and test data samples $\mathbf{x}_i$ $(i = 1, \ldots, n)$ and $\widetilde{\mathbf{x}}_i$ $(i = 1, \ldots, m)$ are each standardised according to:

$$\mathbf{x}_i = \left(\mathbf{x}_i - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\right) \circ \left[\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_d}\right] \tag{B.5}$$

and

$$\widetilde{\mathbf{x}}_i = \left(\widetilde{\mathbf{x}}_i - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\right) \circ \left[\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_d}\right] \tag{B.6}$$

respectively, where $\circ$ denotes the Hadamard product, and σ_i $(i = 1, \ldots, d)$ corresponds to the standard deviation of the ith dimension of the training set $\mathbf{X}$, where it is computed according to:

$$\sigma_i = \sqrt{\frac{1}{n-1}\sum_{p=1}^{n}\left(x_{pi} - \frac{1}{n}\sum_{q=1}^{n}x_{qi}\right)^2} \tag{B.7}$$

According to the previous works in [14], the above listed three pre-processing methods can be formulated in matrix notation, and they are shown in Table B.1.

B.2 Dimensionality Reduction

When the dimensionality of a set of data samples is high, just using simple normalisation techniques may not be sufficient enough to remove the present irrelevant and/or redundant information. Additionally, the high dimensionality can potentially increase the computation complexity of later processing stages. As a result, an additional step is

Table B.1: Listed simple normalisation techniques presented in matrix notation, given the training set $\mathbf{X}$ and the corresponding test set $\widetilde{\mathbf{X}}$. The term $\mathbf{I}_{n\times n}$ is an $n \times n$ identity matrix, and the term $\mathbf{1}_{n\times n}$ is an $n \times n$ matrix with all its elements being 1.

Method	Training Set	Test Set
Rescaling	$\mathbf{S}^{-1}\mathbf{X}$	$\widetilde{\mathbf{S}}^{-1}\widetilde{\mathbf{X}}$
	$\mathbf{S} = \text{diag}\left[\sqrt{\text{diag}(\mathbf{X}\mathbf{X}^T)}\right]$	$\widetilde{\mathbf{S}} = \text{diag}\left[\sqrt{\text{diag}(\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T)}\right]$
Centring	$\left(\mathbf{I}_{n\times n} - \frac{1}{n}\mathbf{1}_{n\times n}\right)\mathbf{X}$	$\widetilde{\mathbf{X}} - \frac{1}{n}\mathbf{1}_{m\times n}\mathbf{X}$
Standardisation	$\left(\mathbf{I}_{n\times n} - \frac{1}{n}\mathbf{1}_{n\times n}\right)\mathbf{X}\mathbf{\Sigma}^{-1}$	$\left(\widetilde{\mathbf{X}} - \frac{1}{n}\mathbf{1}_{m\times n}\mathbf{X}\right)\mathbf{\Sigma}^{-1}$
	$\mathbf{\Sigma} = \text{diag}\left[\sqrt{\frac{1}{n-1}\text{diag}\left[\mathbf{X}^T\left(\mathbf{I}_{n\times n} - \frac{1}{n}\mathbf{1}_{n\times n}\right)\mathbf{X}\right]}\right]$	

needed, which reduces the high dimensionality to a reasonable low one, whilst tending to preserve the meaningful data structures contained within the set and eliminate the present irrelevant and/or redundant information. In practice, the resulting data samples after dimensionality reduction are widely relied by many existing techniques on machine learning and text mining to obtain good learning results [263]. Here, two popular methods in the literature are listed, one is PCA (Principal Component Analysis) and the other one is whitening.

PCA

It first centres a set of data samples, and then does the eigendecomposition of the covariance matrix corresponding to the centred set of data samples to get a $d \times k$ transformation matrix $\mathbf{M}_k$, whose columns are k eigenvectors corresponding to the k largest nonzero eigenvalues. The number of k should be chosen so that over 99% of the total variance is retained in the transformed data samples [263]. The technique achieves dimensionality reduction by eliminating spaces of low variance, and additionally decorrelates the resulting data samples by making the corresponding covariance matrix to be a diagonal matrix. Then, the training and test sets $\mathbf{X}$ and $\widetilde{\mathbf{X}}$ are each transformed according to:

$$\mathbf{X} = \left(\mathbf{I}_{n\times n} - \frac{1}{n}\mathbf{1}_{n\times n}\right)\mathbf{X}\mathbf{M}_k \tag{B.8}$$

and

$$\widetilde{\mathbf{X}} = \left(\widetilde{\mathbf{X}} - \frac{1}{n}\mathbf{1}_{m\times n}\mathbf{X}\right)\mathbf{M}_k \tag{B.9}$$

respectively.

Whitening

It works very similarly to PCA, but additionally rescales the transformed set of data samples along each dimension, so that the corresponding covariance matrix transforms from a diagonal matrix to an identity matrix. Then, the training and test sets $\mathbf{X}$ and $\widetilde{\mathbf{X}}$ are each transformed according to:

$$\mathbf{X} = \left(\mathbf{I}_{n\times n} - \frac{1}{n}\mathbf{1}_{n\times n}\right)\mathbf{X}\mathbf{M}_k\mathbf{\Sigma}_k^{-1} \tag{B.10}$$

and

$$\widetilde{\mathbf{X}} = \left(\widetilde{\mathbf{X}} - \frac{1}{n}\mathbf{1}_{m\times n}\mathbf{X}\right)\mathbf{M}_k\mathbf{\Sigma}_k^{-1} \tag{B.11}$$

respectively, where $\mathbf{\Sigma}_k$ is a $k \times k$ diagonal matrix, with its diagonal element being the corresponding k largest eigenvalues obtained by the eigendecomposition.

Bibliography

[1] Q. Li and H. Ji. Multimodality image registration using local linear embedding and hybrid entropy. *Neurocomputing*, 111:34–42, 2013.

[2] C. Wachinger and N. Navab. Manifold learning for multi-modal image registration. In *British Machine Vision Conference, BMVC 2010, Aberystwyth, UK, August 31 - September 3, 2010. Proceedings*, pages 1–12, 2010.

[3] P. Aljabar, R. Wolz, and D. Rueckert. *Machine Learning in Computer-Aided Diagnosis: Medical Imaging Intelligence and Analysis*, chapter Manifold Learning for Medical Image Registration, Segmentation, and Classification. IGI Global, 2012.

[4] J. Ren, J. Zabalza, S. Marshall, and J. Zheng. Effective feature extraction and data reduction in remote sensing using hyperspectral imaging [applications corner]. *IEEE Trans. Signal Process.*, 31(4):149–154, 2014.

[5] T. Kelman, J. Ren, and S. Marshall. Effective classification of chinese tea samples in hyperspectral imaging. *Artif. Intell. Research*, 2(4):87–96, 2013.

[6] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

[7] B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 2008.

[8] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

[9] I. Steinwart and A. Christmann. *Support Vector Machines*. Springer Science, Business Media, 2008.

[10] N.S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *J AMER STAT*, 46(3):175–185, 1992.

[11] G. Shakhnarovich, T. Darrell, and P. Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press, 2006.

[12] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[13] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[14] T. Mu, J. Jiang, Y. Wang, and J.Y. Goulermas. Adaptive data embedding framework for multiclass classification. *IEEE Trans. Neural Netw.*, 23(8):1291–1303, 2012.

[15] T. Mu, J.Y. Goulermas, J. Tsujii, and S. Ananiadou. Proximity-based frameworks for generating embeddings from multi-output data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2216–2232, 2012.

[16] L. van der Maaten, E. Postma, and J. van den Herik. Dimensionality reduction: A comparative review. Technical Report TiCC TR 2009-005, Tilburg centre for Creative Computing, Tilburg University, 2009.

[17] L.G. Brown. A survey of image registration techniques. *ACM Comput. Surv.*, 24(4):325–376, 1992.

[18] B. Zitová and J. Flusser. Image registration methods: a survey. *Image Vision Comput.*, 21(11):977–1000, 2003.

[19] Z. Anthony, A. Gee, and M. Taylor. Document mosaicing. *Image Vision Comput.*, 17(8):589–595, 1999.

[20] A.A. Goshtasby. *2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications*. John Wiley, Sons, 2005.

[21] A. Sotiras, C. Davatzikos, and N. Paragios. Deformable medical image registration: A survey. *IEEE Trans. Med. Imag.*, 32(7):1153–1190, 2013.

[22] K.M. Simonson, S.M. Drescher, and F.R. Tanner. A statistics-based approach to binary image registration with uncertainty analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):112–125, 2007.

[23] M.J. Westoby, J. Brasington, N.F. Glasser, M.J. Hambrey, and J.M. Reynolds. structure-from-motion photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300–314, 2012.

[24] N. Snavely, S.M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[25] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment - A modern synthesis. In *Vision Algorithms: Theory and Practice, International Workshop on Vision Algorithms, held during ICCV '99, Corfu, Greece, September 21-22, 1999, Proceedings*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer, 1999.

[26] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2003.

[27] P. Clark and R. Boswell. Rule induction with CN2: some recent improvements. In *Machine Learning - EWSL-91, European Working Session on Learning, Porto, Portugal, March 6-8, 1991, Proceedings*, pages 151–163, 1991.

[28] R. Anand, K. Mehrotra, C.K. Mohan, and S. Ranka. Efficient classification for multiclass problems using modular neural networks. *IEEE Trans. Neural Netw.*, 6(1):117–124, 1995.

[29] J.F. Ralph and N.G. Stocks. Fusion of low bit-depth images for battle damage indication. In *13th Conference on Information Fusion, FUSION 2010, Edinburgh, UK, July 26-29, 2010*, pages 1–6. IEEE, 2010.

[30] Y. Chi, E.J. Griffith, and J.F. Ralph. Low bit depth images for small, micro-scale UAVs. In *6th International Conference on Imaging for Crime Detection and Prevention (ICDP 2015), London, UK, 15-17 Jul. 2015*, pages 21–26. IET, 2015.

[31] Y. Chi, E.J. Griffith, J.Y. Goulermas, and J.F. Ralph. Binary data embedding framework for multiclass classification. *IEEE Trans. Human-Mach. Syst.*, 45(4):453–464, 2015.

[32] Y. Chi, E.J. Griffith, J.Y. Goulermas, and J.F. Ralph. Binary data embedding framework for face recognition. In *5th International Conference on Imaging for Crime Detection and Prevention (ICDP 2013), London, UK, 16-17 Dec. 2013*, pages 102–107. IET, 2013.

[33] E.J. Griffith, Y. Chi, M. Jump, and J.F. Ralph. Equivalence of BRISK descriptors for the registration of variable bit-depth aerial imagery. In *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Manchester, UK, 13-16 Oct. 2013*, pages 2587–2592. IEEE, 2013.

[34] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[35] J. Wang and Z. Zhang. Nonlinear embedding preserving multiple local-linearities. *Pattern Recognition*, 43(4):1257–1268, 2010.

[36] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[37] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[38] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[39] W.S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[40] Y. Hou, P. Zhang, X. Xu, X. Zhang, and W. Li. Nonlinear dimensionality reduction by locally linear inlaying. *IEEE Trans. Neural Netw.*, 20(2):300–315, 2009.

[41] J. Wei, Z. Chen, P. Niu, Y. Chen, and W. Chen. Manifold ranking-based locality preserving projections. In *Artificial Intelligence and Computational Intelligence - Third International Conference, AICI 2011, Taiyuan, China, September 24-25, 2011, Proceedings, Part II*, volume 7003 of *Lecture Notes in Computer Science*, pages 664–671. Springer, 2011.

[42] I.T. Jolliffe. *Principal Component Analysis*. Springer Science, Business Media, 2002.

[43] S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[44] X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 153–160. MIT Press, 2003.

[45] E. Kokiopoulou and Y. Saad. Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2143–2156, 2007.

[46] P.K. Chan, M.D.F. Schlag, and J.Y. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 13(9):1088–1096, 1994.

[47] Y. Pan, S. Sam Ge, and A. Al Mamun. Weighted locally linear embedding for dimension reduction. *Pattern Recognition*, 42(5):798–811, 2009.

[48] C. Hou, C. Zhang, Y. Wu, and Y. Jiao. Stable local dimensionality reduction approaches. *Pattern Recognition*, 42(9):2054–2066, 2009.

[49] F.R.K. Chung. *Feature Extraction: Foundations and Applications.* Cbms Regional Conference Series in Mathematics, American Mathematical Soc., 1997.

[50] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugen.*, 7:179–188, 1936.

[51] H. Li, T. Jiang, and K. Zhang. Efficient and robust feature extraction by maximum margin criterion. *IEEE Trans. Neural Netw.*, 17(1):157–165, 2006.

[52] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):40–51, 2007.

[53] T. Zhang, D. Tao, X. Li, and J. Yang. Patch alignment for dimensionality reduction. *IEEE Trans. Knowl. Data Eng.*, 21(9):1299–1313, 2009.

[54] W. Zhang, X. Xue, Z. Sun, Y. Guo, and H. Lu. Optimal dimensionality of metric space for classification. In *ICML*, pages 1135–1142, 2007.

[55] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, 1997.

[56] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(6):607–616, 1996.

[57] M. Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of Machine Learning Research*, 8:1027–1061, 2007.

[58] H. Wang, S. Chen, Z. Hu, and W. Zheng. Locality-preserved maximum information projection. *IEEE Trans. Neural Netw.*, 19(4):571–585, 2008.

[59] S. Ji and J. Ye. Generalized linear discriminant analysis: A unified framework and efficient model selection. *IEEE Trans. Neural Netw.*, 19(10):1768–1782, 2008.

[60] E. Kokiopoulou and Y. Saad. Enhanced graph-based dimensionality reduction with repulsion laplaceans. *Pattern Recognition*, 42(11):2392–2402, 2009.

[61] Q. Gao, H. Xu, Y. Li, and D. Xie. Two-dimensional supervised local similarity and diversity projection. *Pattern Recognition*, 43(10):3359–3363, 2010.

[62] W.K. Wong and H.T. Zhao. Supervised optimal locality preserving projection. *Pattern Recognition*, 45(1):186–197, 2012.

[63] T. Zhang, K. Huang, X. Li, J. Yang, and D. Tao. Discriminative orthogonal neighborhood-preserving projections for classification. *IEEE Trans. Syst., Man, Cybern. B*, 40(1):253–263, 2010.

[64] E. Rodriguez-Martinez, J.Y. Goulermas, T. Mu, and J.F. Ralph. Automatic induction of projection pursuit indices. *IEEE Trans. Neural Netw.*, 21(8):1281–1295, 2010.

[65] S. Zhang. Enhanced supervised locally linear embedding. *Pattern Recognition Letters*, 30(13):1208–1218, 2009.

[66] C. Chen, J. Zhang, and R. Fleischer. Distance approximating dimension reduction of riemannian manifolds. *IEEE Trans. Syst., Man, Cybern. B*, 40(1):208–217, 2010.

[67] Y. Zhang and D. Yeung. Semisupervised generalized discriminant analysis. *IEEE Trans. Neural Netw.*, 22(8):1207–1217, 2011.

[68] Y. Song, F. Nie, C. Zhang, and S. Xiang. A unified framework for semi-supervised dimensionality reduction. *Pattern Recognition*, 41(9):2789–2799, 2008.

[69] M. Sugiyama, T. Idé, S. Nakajima, and J. Sese. Semi-supervised local fisher discriminant analysis for dimensionality reduction. *Machine Learning*, 78(1-2):35–61, 2010.

[70] R. Chatpatanasiri and B. Kijsirikul. A unified semi-supervised dimensionality reduction framework for manifold learning. *Neurocomputing*, 73(10-12):1631–1640, 2010.

[71] G. Chen, Y. Song, F. Wang, and C. Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *Proceedings of the SIAM International Conference on Data Mining,SDM 2008, April 24-26, 2008, Atlanta, Georgia, USA*, pages 410–419, 2008.

[72] Z. Zha, T. Mei, J. Wang, Z. Wang, and X. Hua. Graph-based semi-supervised learning with multiple labels. *J. Visual Communication and Image Representation*, 20(2):97–103, 2009.

[73] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[74] M. Seeger. Learning with labeled and unlabeled data. Technical report, University of Edinburgh, 2001.

[75] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-Supervised Learning Adaptive computation and machine learning*. MIT Press, 2010.

[76] J. Arenas-García, K.B. Petersen, and L.K. Hansen. Sparse kernel orthonormalized pls for feature extraction in large data sets. In *NIPS*, pages 33–40, 2006.

[77] H. Wold. Partial least squares. In *S. Kotz and N. L. Johnson (Eds.)*, volume 6, pages 81–59, 1985.

[78] S. Roweis and C. Brody. Linear heteroencoders. Technical report, Gatsby Computational Neuroscience Unit, University College London, 1999.

[79] D.R. Hardoon, S. Szedmák, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[80] L. Sun, S. Ji, and J. Ye. A least squares formulation for a class of generalized eigenvalue problems in machine learning. In *ICML*, pages 977–984, 2009.

[81] F.R. Bach and M.I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[82] L. Sun, S. Ji, and J. Ye. Hypergraph spectral learning for multi-label classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 668–676, 2008.

[83] Y. Zhang and Z. Zhou. Multilabel dimensionality reduction via dependence maximization. *TKDD*, 4(3), 2010.

[84] P. Daniusis and P. Vaitkus. Supervised feature extraction using hilbert-schmidt norms. In *Intelligent Data Engineering and Automated Learning - IDEAL 2009, 10th International Conference, Burgos, Spain, September 23-26, 2009. Proceedings*, pages 25–33, 2009.

[85] A. Gretton, O. Bousquet, A.J. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic Learning Theory, 16th International Conference, ALT 2005, Singapore, October 8-11, 2005, Proceedings*, pages 63–77, 2005.

[86] L. Song, A.J. Smola, A. Gretton, K.M. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, pages 823–830, 2007.

[87] D. Zhou, J. Huang, and B. Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 1601–1608, 2006.

[88] S.S. Sapatnekar. Overcoming variations in nanometer-scale technologies. *IEEE J. Emerg. Sel. Topics Circuits Syst.*, 1(1):5–18, 2011.

[89] P.J. Phillips, P.J. Flynn, W.T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W.J. Worek. Overview of the face recognition grand challenge. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 947–954. IEEE Computer Society, 2005.

[90] G.H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970.

[91] Deng Cai and Xiaofei He. Orthogonal locality preserving indexing. In *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, pages 3–10. ACM, 2005.

[92] L.K. Saul and S.T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifold. *Journal of Machine Learning Research*, 4:119–155, 2003.

[93] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 2nd edition, 1990.

[94] J. Cohen. Approximate power and sample size determination for common one-sample and two-sample hypothesis tests. *Educational and Psychological Measurement*, 30:811–831, 1970.

[95] J.S. Rossi. Statistical power of psychological research: what have we gained in 20 years? *J Consult Clin Psychol*, 58(5):646–656, 1990.

[96] B. Schölkopf, A.J. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[97] J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London*, 209:415–446, 1909.

[98] B. Schölkopf, C.J.C. Burges, and A.J. Smola. *Advances in Kernel Methods: Support Vector Learning.* MIT Press, 1999.

[99] K. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. Neural Netw.*, 12(2):181–201, 2001.

[100] V.N. Vapnik. *Statistical learning theory.* Wiley, 1998.

[101] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning.* MIT Press, 2012.

[102] G.W. Stewart. *Matrix Algorithms Volume 2: Eigensystems Volume 2 of Matrix algorithms: Eigensystems.* SIAM, 2001.

[103] I. Borg and J. Lingoes. *Multidimensional Similarity Structure Analysis.* Springer-Verlag New York, 1987.

[104] E. Pekalska and R.P.W. Duin. Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters*, 23(8):943–956, 2002.

[105] E. Pekalska, R.P.W. Duin, and P. Paclík. Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, 39(2):189–208, 2006.

[106] T. Mu and S. Ananiadou. Proximity-based graph embeddings for multi-label classification. In *KDIR*, pages 74–84, 2010.

[107] D. Le Gall. Mpeg: A video compression standard for multimedia applications. *Commun. ACM*, 34(4):46–58, 1991.

[108] M. Irani, S.C. Hsu, and P. Anandan. Video compression using mosaic representations. *Sig. Proc.: Image Comm.*, 7(4-6):529–552, 1995.

[109] M. Lee, W. Chen, C.B. Lin, C. Gu, T. Markoc, S.I. Zabinsky, and R. Szeliski. A layered video object coding system using sprite and affine motion model. *IEEE Trans. Circuits Syst. Video Technol.*, 7(1):130–145, 1997.

[110] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Computer Vision - ECCV'92, Second European Conference on Computer Vision, Santa Margherita Ligure, Italy, May 19-22, 1992, Proceedings*, pages 237–252, 1992.

[111] M. Irani and P. Anandan. Video indexing based on mosaic representations. *Proc. IEEE*, 86(5):905–921, 1998.

[112] R. Kumar, P. Anandan, M. Irani, J. Bergen, and K. Hanna. Representation of scenes from collections of images. In *Proceedings IEEE Workshop on Representation of Visual Scenes*, pages 10–17, 1995.

[113] L. Teodosio and W. Bender. Salient video stills: Content and context preserved. In *Proceedings of the First ACM International Conference on Multimedia '93, Anaheim, CA, USA, August 1-6, 1993.*, pages 39–46, 1993.

[114] C.C. Slama, C. Theurer, and S.W. Henriksen, editors. *Manual of photogrammetry*. American Society of Photogrammetry, 4th edition, 1980.

[115] S.E. Chen. Quicktime VR: an image-based approach to virtual environment navigation. In *Proceeding SIGGRAPH '95 Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 29–38, 1995.

[116] S. Mann and R.W. Picard. Virtual bellows: Constructing high quality stills from video. In *Proceedings 1994 International Conference on Image Processing, Austin, Texas, USA, November 13-16, 1994*, pages 363–367, 1994.

[117] R. Szeliski. Image mosaicing for tele-reality applications. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pages 44–53, 1994.

[118] R. Szeliski. Video mosaics for virtual environments. *IEEE Comput. Graph. Appl.*, 16(2):22–30, 1996.

[119] R. Benosman and S.B. Kang, editors. *Panoramic Vision: Sensors, Theory, and Applications*. Springer Science, Business Media, 2001.

[120] H.S. Sawhney, R.T. Kumar, G. Gendel, J.R. Bergen, D. Dixon, and V. Paragano. Videobrush$^{\mathrm{tm}}$: experiences with consumer video mosaicing. In *Proceedings Fourth IEEE Workshop on Applications of Computer Vision, WACV 1998, October 19-21, 1998, Princeton, New Jersey, USA*, pages 56–62, 1998.

[121] F. Badra, A. Qumsieh, and G. Dudek. Rotation and zooming in image mosaicing. In *Proceedings Fourth IEEE Workshop on Applications of Computer Vision, WACV 1998, October 19-21, 1998, Princeton, New Jersey, USA*, pages 50–55, 1998.

[122] M. Brown and D.G. Lowe. Recognising panoramas. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, pages 1218–1227, 2003.

[123] M. Brown, R. Szeliski, and S.A.J. Winder. Multi-image matching using multi-scale oriented patches. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 510–517, 2005.

[124] D.P. Capel and A. Zisserman. Automatic mosaicing with super-resolution zoom. In *1998 Conference on Computer Vision and Pattern Recognition (CVPR'98), June 23-25, 1998, Santa Barbara, CA, USA*, pages 885–891, 1998.

[125] T. Cham and R. Cipolla. A statistical framework for long-range feature matching in uncalibrated image mosaicing. In *1998 Conference on Computer Vision and*

Pattern Recognition (CVPR'98), June 23-25, 1998, Santa Barbara, CA, USA, pages 442–447. IEEE Computer Society, 1998.

[126] P.F. McLauchlan and A. Jaenicke. Image mosaicing using sequential bundle adjustment. *Image Vision Comput.*, 20(9-10):751–759, 2002.

[127] I. Zoghlami, O.D. Faugeras, and R. Deriche. Using geometric corners to build a 2d mosaic from a set of image. In *1997 Conference on Computer Vision and Pattern Recognition (CVPR'97), June 17-19, 1997, San Juan, Puerto Rico*, pages 420–425. IEEE Computer Society, 1997.

[128] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000.

[129] M. Young. Pinhole optics. *Applied Optics*, 10:2763–2767, 1971.

[130] M. Young. The pinhole camera: Imaging without lenses or mirrors. *The Physics Teacher*, 27:648–655, 1989.

[131] O. Faugeras. *Three-dimensional Computer Vision: A Geometric Viewpoint Artificial intelligence MIT Press series in artificial intelligence*. MIT Press, 1993.

[132] R. Szeliski. *Computer Vision: Algorithms and Applications Texts in Computer Science*. Springer Science, Business Media, 2010.

[133] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach Always learning*. Pearson, 2012.

[134] T. Thormählen and H. Broszio. Automatic line-based estimation of radial lens distortion. *Integrated Computer-Aided Engineering*, 12(2):177–190, 2005.

[135] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1), 2006.

[136] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Trans. Robot. Autom.*, 3(4):323–344, 1987.

[137] J. Weng, P.R. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(10):965–980, 1992.

[138] D.C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[139] L.M.G. Fonseca and B.S. Manjunath. Registration techniques for multisensor remotely sensed imagery. *Photogrammetric Engineering, Remote Sensing*, 62(9):1049–1056, 1996.

[140] R.J. Althof, M.G.J. Wind, and J.T. Dobbins. A rapid and automatic image registration algorithms with subpixel accuracy. *IEEE Trans. Med. Imag.*, 16(3):308–316, 1997.

[141] D.I. Barnea and H.F. Silverman. A class of algorithms for fast digital image registration. *IEEE Trans. Comput.*, 21(2):179–186, 1972.

[142] W.K. Pratt. Correlation techniques of image registration. *IEEE Trans. Aerosp. Electron. Syst.*, 10(3):353–358, 1974.

[143] W.K. Pratt. *Digital Image Processing: PIKS Scientific Inside*. Wiley, 4th edition, 2007.

[144] A. Rosenfeld and A.C. Kak. *Digital Picture Processing*. Academic Press, Inc., 2nd edition, 1982.

[145] A.V. Oppenheim, R.W. Schafer, and J.R. Buck. *Discrete-time Signal Processing*. Prentice-Hall, Inc., 2nd edition, 1999.

[146] H. Hanaizumi and S. Fujimur. An automated method for registration of satellite remote sensing images. In *IGARSS*, volume 3, pages 1348–1350, 1993.

[147] R. Berthilsson. Affine correlation. In *International Conference on Pattern Recognition*, pages 1458–1460, 1998.

[148] P. van Wie and M. Stein. A landsat digital image rectification system. *IEEE Trans. Geosci. Electron.*, 15(3):130–137, 1977.

[149] P.E. Anuta. Spatial registration of multispectral and multitemporal digital imagery using fast fourier transform techniques. *IEEE Trans. Geosci. Electron.*, 8(4):353–368, 1970.

[150] E. De Castro and C. Morandi. Registration of translated and rotated images using finite fourier transforms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):700–703, 1987.

[151] Q. Chen, M. Defrise, and F. Deconinck. Symmetric phase-only matched filtering of fourier-mellin transforms for image registration and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(12):1156–1168, 1994.

[152] B.S. Reddy and B.N. Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE Trans. Image Process.*, 5(8):1266–1271, 1996.

[153] Q. Tian and M.N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing*, 35:220–233, 1986.

[154] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81), Vancouver, BC, Canada, August 1981*, pages 674–679, 1981.

[155] G.H. Golub and C.F. van Loan. *Matrix Computations*. JHU Press, 1996.

[156] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[157] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(10):1025–1039, 1998.

[158] J. Flusser and T. Suk. A moment-based approach to registration of images with affine geometric distortion. *IEEE Trans. Geosci. Remote Sens.*, 32(2):382–387, 1994.

[159] A.A. Goshtasby, G.C. Stockman, and C.V. Page. A region-based approach to digital image registration with subpixel accuracy. *IEEE Trans. Geosci. Remote Sens.*, 24(3):390–399, 1986.

[160] A.A. Goshtasby and G.C. Stockman. Point pattern matching using convex hull edges. *IEEE Trans. Syst., Man, Cybern.*, 15(5):631–637, 1985.

[161] M. Helm. Towards automatic rectification of satellite images using feature based matching. In *International Geoscience and Remote Sensing Symposium IGARSS 91*, volume 4, pages 2439–2442, 1991.

[162] Y.C. Hsieh, D.M. McKeown, and F.P. Perlant. Performance evaluation of scene registration and stereo matching for cartographic feature extraction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):214–238, 1992.

[163] S.N. Shore. *Forces in Physics: A Historical Perspective*. Greenwood Press, 2008.

[164] N.R. Pal and S.K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.

[165] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[166] E. Dougherty. *Mathematical Morphology in Image Processing*. CRC Press, 1992.

[167] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Syst., Man, Cybern.*, 9(1):62–66, 1979.

[168] S. Moss and E.R. Hancock. Multiple line-template matching with the em algorithm. *Pattern Recognition Letters*, 18:1283–1292, 1997.

[169] W.H. Wang and Y.C. Chen. Image registration by control points pairing using the invariant properties of line segments. *Pattern Recognition Letters*, 18(3):269–281, 1997.

[170] X. Dai and S. Khorram. Development of a feature-based approach to automated image registration for multitemporal and multisensor remotely sensed imagery. In *IGARSS*, volume 1, pages 243–245, 1997.

[171] V. Govindu, C. Shekhar, and R. Chellappa. Using geometric properties for correspondence-less image alignment. In *ICPR*, volume 1, pages 37–41, 1998.

[172] H. Li, B.S. Manjunath, and S.K. Mitra. A contour-based approach to multisensor image registration. *IEEE Trans. Image Process.*, 4(3):320–334, 1995.

[173] S.Z. Li, J. Kittler, and M. Petrou. Matching and recognition of road networks from aerial images. In *Computer Vision - ECCV 92*, volume 588, pages 857–861, 1992.

[174] N. Vujovic and D. Brzakovic. Establishing the correspondence between control points in pairs of mammographic images. *IEEE Trans. Image Process.*, 6(10):1388–1399, 1997.

[175] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[176] D. Marr and E. Hildreth. Theory of edge detection. In *Proceedings of the Royal Society of London*, pages 187–217, 1980.

[177] G.C. Stockman, S. Kopstein, and S. Benett. Matching images to models for registration and object detection via clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 4(3):229–241, 1982.

[178] A.S. Vasileisky, B. Zhukov, and M. Berger. Automated image coregistration based on linear feature recognition. In *Proceedings of the Second Conference Fusion of Earth Data, Sophia Antipolis, France*, page 5966, 1998.

[179] M. Ehlers and M. Fuller. Region-based matching for image integration in remote sensing databases. In *IGARSS*, volume 4, pages 2231–2234, 1991.

[180] B.S. Manjunath, C. Shekhar, and R. Chellappa. A new approach to image feature detection with applications. *Pattern Recognition*, 29(4):627–640, 1996.

[181] Q. Zheng and R. Chellappa. A computational vision approach to image registration. *IEEE Trans. Image Process.*, 2(3):311–326, 1993.

[182] W.S.I. Ali and F.S. Cohen. Registering coronal histological 2-d sections of a rat brain with coronal sections of a 3-d brain atlas using geometric curve invariants and b-spline representation. *IEEE Trans. Med. Imag.*, 17(6):957–966, 1998.

[183] S. Banerjee, D.P. Mukherjee, and D.D. Majumdar. Point landmarks for registration of ct and mr images. *Pattern Recognition Letters*, 16:1033–1042, 1995.

[184] L.M.G. Fonseca and M.H.M. Costa. Automatic registration of satellite images. In *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, pages 219–226, 1997.

[185] J.W. Hsieh, H.Y.M. Liao, K.C. Fam, and M.T. Ko. A fast algorithm for image registration without predetermining correspondences. In *ICPR*, pages 765–769, 1996.

[186] D. Bhattacharya and S. Sinha. Invariance of stereo images via the theory of complex moments. *Pattern Recognition*, 30(9):1373–1386, 1997.

[187] C. Wang, H. Sun, S. Yada, and A. Rosenfeld. Some experiments in relaxation image matching using corner features. *Pattern Recognition*, 16(2):167–182, 1983.

[188] K. Rohr. Localization properties of direct corner detectors. *Journal of Mathematical Imaging and Vision*, 4(2):139–150, 1994.

[189] K. Rohr. *Landmark-Based Image Analysis: Using Geometric and Intensity Models*. Springer Science, Business Media, 2001.

[190] Z. Zheng, H. Wang, and E.K. Teoh. Analysis of gray level corner detection. *Pattern Recognition Letters*, 20(2):149–162, 1999.

[191] D.G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[192] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[193] H. Bay, T. Tuytelaars, and L.J. Van Gool. SURF: speeded up robust features. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer, 2006.

[194] H. Bay, A. Ess, T. Tuytelaars, and L.J. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[195] G. Carneiro and A.D. Jepson. The distinctiveness, detectability, and robustness of local image features. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 296–301. IEEE Computer Society, 2005.

[196] C.S. Kenney, M. Zuliani, and B.S. Manjunath. An axiomatic approach to corner detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 191–197. IEEE Computer Society, 2005.

[197] B. Platel, E. Balmachnova, L. Florack, F. Kanters, and B.M. ter Haar Romeny. Using top-points as interest points for image matching. In *Deep Structure, Singularities, and Computer Vision, First International Workshop, DSSCV 2005, Maastricht, The Netherlands, June 9-10, 2005, Revised Selected Papers*, volume 3753 of *Lecture Notes in Computer Science*, pages 211–222. Springer, 2005.

[198] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*, volume 3951 of *Lecture Notes in Computer Science*, pages 430–443. Springer, 2006.

[199] S. Leutenegger, M. Chli, and R. Siegwart. BRISK: binary robust invariant scalable keypoints. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2548–2555. IEEE, 2011.

[200] H.P. Moravec. The stanford cart and the cmu rover. *Proc. IEEE*, 71(7):872–884, 1983.

[201] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, 1989.

[202] J. Shi and C. Tomasi. Good features to track. In *Conference on Computer Vision and Pattern Recognition, CVPR 1994, 21-23 June, 1994, Seattle, WA, USA*, pages 593–600. IEEE, 1994.

[203] W. Förstner. A feature based correspondence algorithm for image matching. *Int. Arch. of Photogrammetry and Remote Sensing*, 26(3):150–166, 1986.

[204] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference, AVC 1988, Manchester, UK, September, 1988*, pages 1–6. Alvey Vision Club, 1988.

[205] W. Förstner. A framework for low level feature extraction. In *Computer Vision - ECCV'94, Third European Conference on Computer Vision, Stockholm, Sweden, May 2-6, 1994, Proceedings, Volume II*, volume 801 of *Lecture Notes in Computer Science*, pages 383–394. Springer, 1994.

[206] B. Triggs. Detecting keypoints with stable position, orientation, and scale under illumination changes. In *Computer Vision - ECCV 2004, 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part IV*, volume 3024 of *Lecture Notes in Computer Science*, pages 100–113. Springer, 2004.

[207] T. Lindeberg. Scale-space for discrete signals. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(3):234–254, 1990.

[208] A.P. Witkin. Scale-space filtering. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence. Karlsruhe, FRG, August 1983*, pages 1019–1022. William Kaufmann, 1983.

[209] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270, 1994.

[210] M. Felsberg and G. Sommer. The monogenic scale-space: A unifying approach to phase-based image processing in scale-space. *Journal of Mathematical Imaging and Vision*, 21(1-2):5–26, 2004.

[211] L.G. Shapiro and G.C. Stockman. *Computer Vision*. Prentice Hall, 2001.

[212] M. Nixon. *Feature Extraction and Image Processing*. Academic Press, 2nd edition, 2008.

[213] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part I*, volume 2350 of *Lecture Notes in Computer Science*, pages 128–142. Springer, 2002.

[214] A. Baumberg. Reliable feature matching across widely separated views. In *2000 Conference on Computer Vision and Pattern Recognition (CVPR 2000), 13-15 June 2000, Hilton Head, SC, USA*, pages 1774–1781. IEEE Computer Society, 2000.

[215] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *ICCV*, pages 525–531, 2001.

[216] M. Brown and D.G. Lowe. Invariant features from interest point groups. In *Proceedings of the British Machine Vision Conference 2002, BMVC 2002, Cardiff, UK, 2-5 September 2002*, pages 1–10. British Machine Vision Association, 2002.

[217] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR (2)*, pages 506–513, 2004.

[218] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence. Cambridge, MA, August 1977*, pages 659–663. William Kaufmann, 1977.

[219] S. Ranade and A. Rosenfeld. Point pattern matching by relaxation. *Pattern Recognition*, 12(4):269–275, 1980.

[220] S.A. Nene and S.K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(9):989–1003, 1997.

[221] J.S. Beis and D.G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), June 17-19, 1997, San Juan, Puerto Rico*, pages 1000–1006. IEEE Computer Society, 1997.

[222] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley Publishing Company, 1990.

[223] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[224] S. Choi, T. Kim, and W. Yu. Performance evaluation of RANSAC family. In *British Machine Vision Conference, BMVC 2009, London, UK, September 7-10, 2009. Proceedings*, pages 1–12. British Machine Vision Association, 2009.

[225] C.V. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999.

[226] P.J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79:871–880, 1984.

[227] O. Chum and J. Matas. Matching with PROSAC - progressive sample consensus. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 220–226. IEEE Computer Society, 2005.

[228] S.M. Stigler. Gauss and the invention of least squares. *Ann. Statist.*, 9(3):465–474, 1981.

[229] O. Bretscher. *Linear Algebra with Applications*. Prentice Hall, 2nd edition, 2001.

[230] T. Tuytelaars and L.J. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[231] P. Hellier and C. Barillot. Coupling dense and landmark-based approaches for non rigid registration. *IEEE Trans. Med. Imag.*, 22(2):217–227, 2003.

[232] M. Hazewinkel. *Encyclopaedia of Mathematics*. Springer, 1987.

[233] N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, 2006.

[234] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, 1998.

[235] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593, 1997.

[236] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Science, Business Media, 1999.

[237] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[238] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004.

[239] R.I. Hartley and P. Sturrn. Triangulation. In *American Image Understanding Workshop*, pages 957–966, 1994.

[240] P.H.S. Torr, A.W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*, 32(1):27–44, 1999.

[241] P.H.S. Torr, A. Zisserman, and S.J. Maybank. Robust detection of degenerate configurations for the fundamental matrix. In *ICCV*, pages 1037–1042, 1995.

[242] F. Schaffalitzky, A. Zisserman, R.I. Hartley, and P.H.S. Torr. A six point solution for structure and motion. In *Computer Vision - ECCV 2000, 6th European Conference on Computer Vision, Dublin, Ireland, June 26 - July 1, 2000, Proceedings, Part I*, volume 1842 of *Lecture Notes in Computer Science*, pages 632–648. Springer, 2000.

[243] R.I. Hartley. In defence of the 8-point algorithm. In *ICCV*, pages 1064–1070, 1995.

[244] P.H.S. Torr and A. Zisserman. Robust computation and parametrization of multiple view relations. In *ICCV*, pages 727–732, 1998.

[245] Z. Zhang, R. Deriche, O.D. Faugeras, and Q. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artif. Intell.*, 78(1-2):87–119, 1995.

[246] Y. Zheng, S. Sugimoto, and M. Okutomi. A practical rank-constrained eight-point algorithm for fundamental matrix estimation. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 1546–1553. IEEE, 2013.

[247] T.S. Huang and A.N. Netravali. Motion and structure from feature correspondences: a review. *Proc. IEEE*, 82(2):252–268, 1994.

[248] R.I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *Computer Vision - ECCV'92, Second European Conference on Computer Vision, Santa Margherita Ligure, Italy, May 19-22, 1992, Proceedings*, volume 588 of *Lecture Notes in Computer Science*, pages 579–587. Springer, 1992.

[249] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.

[250] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*. IEEE Computer Society, 2007.

[251] Y. Furukawa, B. Curless, S.M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 1434–1441. IEEE Computer Society, 2010.

[252] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M.F. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part II*, volume 3952 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2006.

[253] H. Zhou, A. Mian, L. Wei, D. Creighton, M. Hossny, and S. Nahavandi. Recent advances on singlemodal and multimodal face recognition: A survey. *IEEE Trans. Hum.-Mach. Syst.*, 44(6):701–716, 2014.

[254] A.F. Abate, M. Nappi, D. Riccio, and G. Sabatino. 2d and 3d face recognition: A survey. *Pattern Recognition Letters*, 28(14):1885–1906, 2007.

[255] C. Xu, Y. Wang, T. Tan, and L. Quan. Depth vs. intensity: Which is more important for face recognition? In *17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, August 23-26, 2004.*, pages 342–345, 2004.

[256] S.G. Kong, J. Heo, B.R. Abidi, J. Paik, and M.A. Abidi. Recent advances in visual and infrared face recognition - a review. *Computer Vision and Image Understanding*, 97(1):103–135, 2005.

[257] G. Bebis, A. Gyaourova, S. Singh, and I. Pavlidis. Face recognition by fusing thermal infrared and visible imagery. *Image Vision Comput.*, 24(7):727–742, 2006.

[258] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004.

[259] J. Fan and Y. Fan. High dimensional classification using features annealed independence rules. *Ann. Statist.*, 36(6):2605–2637, 2008.

[260] W. You, Z. Yang, M. Yuan, and G. Ji. Totalpls: Local dimension reduction for multicategory microarray data. *IEEE Trans. Hum.-Mach. Syst.*, 44(1):125–138, 2014.

[261] L. Sun, A. Korhonen, and Y. Krymolowski. Automatic classification of english verbs using rich syntactic features. In *Third International Joint Conference on Natural Language Processing, IJCNLP 2008, Hyderabad, India, January 7-12, 2008*, pages 769–774, 2008.

[262] J. Bjorne, J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, and T. Salakoski. Extracting contextualized complex biological events with rich graph-based feature sets. *Computational Intelligence*, pages 541–557, 2011.

[263] A. Ng, J. Ngiam, C. Foo, Y. Mai, and C. Suen. Data preprocessing. `http://ufldl.stanford.edu/wiki/index.php/Data_Preprocessing`, Apr 2013. UFLDL (Unsupervised Feature Learning and Deep Learning) Tutorial.

[264] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(1):4–37, 2000.

[265] E. Kokiopoulou, J. Chen, and Y. Saad. Trace optimization and eigenproblems in dimension reduction methods. *NLAA*, 18:565–602, 2011.

[266] H. Wang, S. Yan, D. Xu, X. Tang, and T.S. Huang. Trace ratio vs. ratio trace for dimensionality reduction. In *CVPR*, 2007.

[267] Y. Jia, F. Nie, and C. Zhang. Trace ratio problem revisited. *IEEE Trans. Neural Netw.*, 20(4):729–735, 2009.

[268] A.C. Lorena, A.C. de Carvalho, and J. Gama. A review on the combination of binary classifiers in multiclass problems. *Artif. Intell. Rev.*, 30(1-4):19–37, 2008.

[269] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 44(8):1761–1776, 2011.

[270] C. Hsu and C. Lin. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.*, 13(2):415–425, 2002.

[271] N.V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explorations*, 6(11):1–6, 2004.

[272] H. He and E.A. Garcia. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.*, 21(9):1263–1284, 2009.

[273] Y. Sun, A. Wong, and M.S. Kamel. Classification of imbalanced data: a review. *IJPRAI*, 23(4):687–719, 2009.

[274] R. Yan, J. Zhang, J. Yang, and A.G. Hauptmann. A discriminative learning framework with pairwise constraints for video object classification. In *CVPR (2)*, pages 284–291, 2004.

[275] T. Mu, A.K. Nandi, and R.M. Rangayyan. Analysis of breast tumors in mammograms using the pairwise rayleigh quotient classifier. *J. Electronic Imaging*, 16(4):043004, 2007.

[276] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision, Sarasota FL*, 1994.

[277] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database. In *FGR*, pages 53–58, 2002.

[278] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):643–660, 2001.

[279] K. Lee, J. Ho, and D.J. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(5):684–698, 2005.

[280] D. Cai, X. He, W.V. Zhang, and J. Han. Regularized locality preserving indexing via spectral regression. In *CIKM*, pages 741–750, 2007.

[281] D. Cai, X. He, and J. Han. Locally consistent concept factorization for document clustering. *IEEE Trans. Knowl. Data Eng.*, 23(6):902–913, 2011.

[282] E. Alpaydin. *Introduction to Machine Learning*. MIT Press, 2nd edition, 2010.

[283] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 2000.

[284] A. Gaszczak, T.P. Breckon, and J.W. Han. Real-time people and vehicle detection from uav imagery. In *Proc. SPIE Conference Intelligent Robots and Computer Vision XXVIII: Algorithms and Techniques*, volume 7878, 2011.

[285] K.B. Sandvik and K. Lohne. The rise of the humanitarian drone: Giving content to an emerging concept. *Millennium - Journal of International Studies*, 43(1):145–164, 2014.

[286] T. Wall and T. Monahan. Surveillance and violence from afar: The politics of drones and liminal security-scapes. *Theoretical Criminology*, 15(3):239–254, 2011.

[287] S. Yahyanejad, M. Quaritsch, and B. Rinner. Incremental, orthorectified and loop-independent mosaicking of aerial images taken by micro uavs. In *2011 IEEE International Symposium on Robotic and Sensors Environments, ROSE 2012, Montréal, Canada, September 17-18, 2011*, pages 137–142. IEEE, 2011.

[288] K.L. Edwards. Air-to-ground targeting - UAVs, data links and interoperability (project extendor). *The Aeronautical Journal of the Royal Aeronautical Society*, 108(1088):493–504, 2004.

[289] G. Wade. *Signal Coding and Processing*. Cambridge University Press, 1994.

[290] A. Çelebi, O. Urhan, I. Hamzaoglu, and S. Ertürk. Efficient hardware implementations of low bit depth motion estimation algorithms. *IEEE Signal Process. Lett.*, 16(6):513–516, 2009.

[291] N. Kim, S. Ertürk, and H. Lee. Two-bit transform based block motion estimation using second derivatives. *IEEE Trans. Consum. Electron.*, 55(2):902–910, 2009.

[292] A. Vlachos, V.E. Fotopoulos, and A.N. Skodras. Low bit depth representation motion estimation algorithms: a comparative study. *J. Real-Time Image Processing*, 5(3):141–148, 2010.

[293] S. Yahyanejad, D. Wischounig-Strucl, M. Quaritsch, and B. Rinner. Incremental mosaicking of images from autonomous, small-scale uavs. In *Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2010, Boston, MA, USA, August 29 - September 1, 2010*, pages 329–336. IEEE Computer Society, 2010.

[294] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

[295] R. Dosselmann and X.D. Yang. A comprehensive assessment of the structural similarity index. *Signal, Image and Video Processing*, 5(1):81–91, 2011.

[296] Z. Wang, L. Lu, and A.C. Bovik. Video quality assessment based on structural distortion measurement. *Sig. Proc.: Image Comm.*, 19(2):121–132, 2004.

[297] R. Hodge, J. Brasington, and K. Richards. In situ characterization of grain-scale fluvial morphology using terrestrial laser scanning. *Earth Surface Processes and Landforms*, 34(7):954–968, 2009.

[298] B. Notebaert, G. Verstraeten, G. Govers, and J. Poesen. Qualitative and quantitative applications of lidar imagery in fluvial geomorphology. *Earth Surface Processes and Landforms*, 34(2):217–231, 2009.

[299] J. Brasington and R.M.A. Smart. Close range digital photogrammetric analysis of experimental drainage basin evolution. *Earth Surface Processes and Landforms*, 28(3):231–247, 2003.

[300] C. Wu. VisualSFM: A visual structure from motion system. `http://ccwu.me/vsfm`, 2015.

[301] Y. Furukawa. Clustering views for multi-view stereo (CMVS). `http://www.di.ens.fr/cmvs`, 2015.

[302] A. Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[303] Y. Goldberg and M. Elhadad. splitsvm: Fast, space-efficient, non-heuristic, polynomial kernel computation for NLP applications. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA, Short Papers*, pages 237–240, 2008.

[304] T.T. Tanimoto. An elementary mathematical theory of classification and prediction. Technical Report 8, Internal IBM Technical Report, 1957.

[305] D.J. Rogers and T.T. Tanimoto. A computer program for classifying plants. *Science*, 132:1115–1118, 1960.

[306] H. Qian, X. Wu, and Y. Xu. *Intelligent Surveillance Systems*. Springer Science, Business Media, 2011.

[307] W. Chojnacki and M.J. Brooks. A note on the locally linear embedding algorithm. *IJPRAI*, 23(8):1739–1752, 2009.